

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ
СІКОРСЬКОГО»**

Теплоенергетичний факультет

Кафедра автоматизації проектування енергетичних процесів і систем

"На правах рукопису"
УДК 004.021

«До захисту допущено»

Завідувач кафедри

_____ О.В. Коваль

(підпис) (ініціали, прізвище)

“ _____ ” _____ 2019 р.

Магістерська дисертація

зі спеціальності 121 Інженерія програмного забезпечення

за спеціалізацією Інженерія програмного забезпечення розподілених систем

на тему _____ “Оптимізація коефіцієнтів передавальної функції цифрового фільтра на основі генетичного алгоритму”

Виконав: студент 6 курсу, групи ТВ-82мп

_____ Віннічук Михайло Володимирович

(прізвище, ім'я, по батькові)

_____ (підпис)

Науковий керівник _____ доц., к.т.н. Варава І.А.

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

_____ (підпис)

Консультант _____

(назва розділу)

_____ (вчені ступінь та звання, прізвище, ініціали)

_____ (підпис)

Рецензент _____

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

_____ (підпис)

Засвідчую, що у цій магістерській дисертації
немає запозичень з праць інших авторів без
відповідних посилань.

Студент _____

(підпис)

Київ – 2019

Національний технічний університет України
“ Київський політехнічний інститут ім. Ігоря Сікорського”

Факультет теплоенергетичний

Кафедра автоматизації проектування енергетичних процесів і систем

Рівень вищої освіти другий, магістерський

зі спеціальності - 121 Інженерія програмного забезпечення

за спеціалізацією - Інженерія програмного забезпечення розподілених систем

ЗАТВЕРДЖУЮ
Завідувач кафедри
Коваль О.В. _____
(прізвище, ініціали)
« ____ » _____ 2019р.

З А В Д А Н Н Я
НА МАГІСТЕРСЬКУ ДИСЕРТАЦІЮ СТУДЕНТУ

Віннічук Михайло Володимирович

(прізвище, ім'я, по батькові)

1. Тема дисертації Оптимізація коефіцієнтів передавальної функції цифрового
фільтра на основі генетичного алгоритму

Науковий керівник доц., к.т.н., Варава І.А.

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від “ 4 ” листопада 2019 року № 3812-с

2. Строк подання студентом дисертації 9 грудня 2019

3. Об'єкт дослідження передавальна функція цифрового фільтра з частотною
вибіркою

4. Предмет дослідження оптимізація коефіцієнтів передавальної функції
цифрового фільтра з частотною вибіркою за допомогою генетичного алгоритму

5. Перелік питань, які потрібно
розробити _____

Аналіз літератури стосовно проектування цифрових фільтрів з оптимальними
параметрами і літератури стосовно методів глобальної оптимізації багатовимірних
функцій; модифікувати генетичний алгоритм для оптимізації коефіцієнтів
передавальної функції цифрового фільтра в перехідній смузі; програмне
забезпечення для проектування цифрових фільтрів на основі частотної вибірки з
оптимізацією коефіцієнтів передавальної функції; сховище спроектованих фільтрів
для збереження оптимальних коефіцієнтів передавальної функції для заданих
параметрів цифрового фільтру

6. Орієнтований перелік ілюстративного матеріалу
«Тема та завдання роботи», «Види цифрових фільтрів», «Методи проектування цифрових фільтрів», «Огляд програмних засобів для проектування цифрових фільтрів», «Розрахункові формули передавальної функції фільтру на основі частотної вибірки», «Побудова графіку функції згасання цифрового фільтру за частотною вибіркою», «Етапи оптимізації», «Етапи генетичного алгоритму», «Багатоекстремальність функції гарантованого згасання», «Реалізація генетичного алгоритму в роботі», «Архітектура програми», «Приклад вводу даних в програму», «Ітерація генетичного алгоритму», «Порівняння простого перебору і генетичного алгоритму», «Збереження результатів», «Висновок»

7. Орієнтований перелік публікацій _____

Оптимізація коефіцієнтів цифрового фільтру методом частотної вибірки за допомогою генетичного алгоритму (Тези на конференцію «VI Всеукраїнська науково-практична конференція молодих науковців (м.Київ, 16 травня 2019р.)»

8. Дата видачі завдання « 28 » _____ вересня _____ 2019р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання магістерської дисертації	Строки виконання етапів магістерської дисертації	Примітка
1	Отримання завдання	28.09.18 р.	
2	Опрацювання літературних джерел	01.10.18 р. – 03.02.19 р.	
3	Підготовка матеріалів дисертації	04.02 – 31.05.19 р.	
4	Підготовка доповідей на конференції	11.03 – 29.03.19 р.	
6	Розробка програмного продукту	03.06 – 25.10.19 р.	
5	Переддипломна практика	02.09 – 25.10.19 р.	
7	Захист програмного продукту	26.10.19 р.	
8	Розробка стартап-проекту	11.11 – 19.11.19 р.	
9	Передзахист	20.11.19 р.	
10	Оформлення дисертації	21.11- 29.11.19 р.	
11	Захист	16.12.19 р.	

Студент

_____ (підпис)

Віннічук М.В.

_____ (прізвище та ініціали)

Науковий керівник

_____ (підпис)

Варава І.А.

_____ (прізвище та ініціали)

РЕФЕРАТ

Структура й обсяг дипломної роботи

Магістерська дисертація складається зі вступу, шести розділів, висновку, переліку посилань з 30 найменувань, 2 додатків і містить 23 рисунки, 22 таблиці. Повний обсяг магістерської дисертації складає 77 сторінок, з яких перелік посилань займає 2 сторінки.

Актуальність теми. Цифрова фільтрація – це те, без чого сьогодні не обходиться ніхто. Такі речі як мобільні телефони, медичні пристрої, розумні транспортні засоби, радарні станції, телекомунікації можуть робити різноманітні речі, такі як програвання аудіо і відео, обробка фотографій, прийом і передача сигналів по комп'ютерній мережі, оцінка власного перебування в просторі – все це було б неможливим без цифрової фільтрації.

Цифрові фільтри задаються такими характеристиками як порядок, частота зрізу, гарантоване затухання, нерівномірність. І щоб забезпечити головну задачу фільтру – виділення корисного сигналу, потрібно забезпечити гарантоване затухання сигналу на частотах зрізу, і максимально прибрати нерівномірність в полосі пропускання. І забезпечити це можливо знайшовши правильний набір коефіцієнтів передавальної функції цифрового фільтру. Функція що оптимізується в даному випадку – це гарантоване затухання. Але вона містить багато екстремумів. Класичним методам оптимізації важко працювати в таких умовах. Проте методи глобальної оптимізації тут допомагають. Одним з таких методів є генетичний алгоритм.

Зв'язок роботи з науковими програмами, планами, темами

Дисертаційна робота магістра виконувалась у КПІ ім. Ігоря Сікорського у відповідності з планом наукових досліджень кафедри АПЕПС.

Мета дослідження. Метою дослідження є створення продукту, що дозволить проектувати цифровий фільтр з оптимальними характеристиками передавальної функції за рахунок генетичного алгоритму.

Для реалізації поставленої мети були сформульовані наступні **завдання дослідження**, що визначили логіку дослідження та його структуру:

- Аналіз літератури стосовно проектування цифрових фільтрів з оптимальними параметрами і літератури стосовно методів глобальної оптимізації багатовимірних функцій;
- Адаптація і реалізація генетичного алгоритму для оптимізації коефіцієнтів передавальної функції цифрового фільтра в полосі пропускання і полосах фільтрування;
- Розробити програмне забезпечення для проектування цифрових фільтрів на основі частотної вибірки з оптимізацією коефіцієнтів передавальної функції;
- . Розробити сховище даних спроектованих фільтрів для збереження оптимальних коефіцієнтів передавальної функції для заданих параметрів цифрового фільтру.

Об'єктом дослідження є передавальна функція цифрового фільтра з частотною вибіркою.

Предмет дослідження є оптимізація коефіцієнтів передавальної функції цифрового фільтра з частотною вибіркою за допомогою генетичного алгоритму.

Методи дослідження: При вирішенні задач роботи застосовувались наступні методи:

- Генетичний алгоритм;
- Частотна вибірка.

Наукова новизна одержаних результатів. Модифікований метод генетичного алгоритму для оптимізації передавальної функції цифрового фільтру.

Практичне значення одержаних результатів роботи полягає в тому що її можна використати для проектування оптимальних цифрових фільтрів з частотною вибіркою.

Ключові слова. *ЧАСТОТНА ВИБІРКА, ГЕНЕТИЧНИЙ АЛГОРИТМ, СИГНАЛ, ЦИФРОВА ФІЛЬТРАЦІЯ.*

ABSTRACT

The structure and volume of the thesis.

The master's thesis consists of an introduction, six sections, conclusion, a list of links of 30 titles, 2 appendices and contains 23 drawings, 22 tables. The full volume of the master's thesis is 77 pages, of which the list of links occupies 2 pages.

Actuality of theme. Digital filtering is something that nobody can do today. Things like mobile phones, medical devices, smart vehicles, radar stations, telecommunications can do a variety of things, such as audio and video playback, photo processing, receiving and transmitting signals over a computer network, evaluating your own space would be impossible without digital filtering.

The digital filters are given such characteristics as order, cutoff frequency, guaranteed damping, non-uniformity. And to provide the main task of the filter - the selection of useful signal, it is necessary to provide a guaranteed attenuation of the signal at the cutoff frequencies, and to remove as much as possible the irregularity in the bandwidth. And to ensure this is possible by finding the right set of coefficients of the transfer function of the digital filter. The optimized function in this case is a guaranteed damping. But it contains many extremes. Classical optimization methods are difficult to work under such conditions. However, global optimization techniques are helpful here. One such method is the genetic algorithm.

Relationship with academic programs, plans, themes.

The dissertation work of the master was carried out in KPI them. Igor Sikorsky in accordance with the plan of scientific research of the Department of APEPPS.

The aim of the study. The purpose of the study is to create a product that will allow the design of a digital filter with the optimal characteristics of the transfer function due to the genetic algorithm.

To achieve this goal the following objectives were formulated research **following tasks** the logic and structure:

- Analysis of literature on design of digital filters with optimal parameters and literature on methods of global optimization of multidimensional functions;

- Adaptation and implementation of a genetic algorithm to optimize the coefficients of the transfer function of a digital filter in the bandwidth and filtering bands;
- Develop software for designing digital filters based on frequency sampling with optimization of transmission coefficients;
- Design a data filter repository to store optimal transfer coefficients for a given digital filter parameter.

Object is a transfer function of a digital filter with frequency sampling.

The subject is the optimization of the coefficients of the transfer function of a digital filter with frequency sampling by means of a genetic algorithm.

Methods: The following methods were used in solving the problems of work:

- Genetic algorithm;
- Frequency sampling.

Scientific novelty of the results. A modified genetic algorithm method to optimize the transfer function of a digital filter.

The practical significance of the results of job is that it can be used to design optimal frequency-sampled digital filters.

Keywords. *FREQUENCY SELECTION, GENETIC ALGORITHM, SIGNAL, DIGITAL FILTRATION.*

ЗМІСТ

Перелік умовних позначень, скорочень і термінів	10
Вступ	11
1 Задача розробки системи оптимізації коефіцієнтів передавальної функції цифрового фільтра за допомогою генетичного алгоритму	13
2 Аналіз проблеми оптимізації коефіцієнтів передавальної функції цифрового фільтра за допомогою генетичного алгоритму	14
2.1 Цифрові фільтри.....	14
2.2 Види цифрових фільтрів	16
2.3 Методи проектування цифрових фільтрів	17
2.4 Актуальність цифрової фільтрації.....	21
2.5 Порівняння існуючих рішень	21
2.6 Сучасні підходи до побудови методів глобальної оптимізації.....	23
2.7 Властивості методів глобальної оптимізації	26
2.8 Методи глобальної оптимізації	28
2.9 Генетичний алгоритм	33
2.10 Результати експериментів під час розробки	34
Висновок до розділу 2.....	34
3 Опис програмної реалізації	36
3.1 Основи генетичного алгоритму	36
3.2 Пошук гарантованого затухання	37
3.3 Знаходження найкращого рішення	39
3.4 Алгоритм роботи генетичного алгоритму	41
3.5 Реалізація оновлення графіка гарантованого затухання	42
3.6 Використання бази даних в програмі.....	43
Висновок до розділу 3.....	44
4 Опис використаних програмних засобів	45

4.1 Мова програмування C#	45
4.2 Технологія Windows Forms	46
4.3 Середовище розробки Visual Studio	47
Висновок до розділу 4.....	47
5 Методика роботи користувача з програмною системою	48
5.1 Системні вимоги та інсталяція	48
5.2 Сценарій роботи користувача з системою	48
Висновок до розділу 5.....	54
6 Розробка стартап проекту	55
6.1 Опис ідеї проекту	55
6.2 Технологічний аудит ідеї проекту.....	58
6.3 Аналіз ринкових можливостей запуску стартап-проекту	59
6.4 Розробка ринкової стратегії проекту.....	66
6.5 Розроблення маркетингової програми	69
Висновок до розділу 6.....	73
Висновки	74
Список використаних джерел	75
Додаток А	78

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ЦФ — цифровий фільтр.

СІХ-фільтр — фільтр із скінченною імпульсною характеристикою.

НІХ-фільтр — фільтр із нескінченною імпульсною характеристикою.

ВСТУП

Наш світ наповнений величезною кількістю різних сигналів. І для того щоб мати змогу його досліджувати, треба вміти працювати з ними. Однією з задач яка стоїть перед нами коли ми маємо змогу отримати якусь інформацію – це виділення корисної нам частини. Виділення корисного сигналу необхідно, тому що сигнал який надходить у систему із зовнішнього середовища, змішаний із шумами, викликаними різноманітними фізичними процесами, що мають, як правило, випадковий характер.

Для виконання даної роботи людство винайшло фільтри: аналогові і цифрові. В межах поточної роботи було розглянуто цифрові фільтри.

Сьогодні такі фільтри застосовуються практично всюди, де потрібна обробка сигналів, зокрема у спектральному аналізі, обробці зображень, обробці відео, обробці мови та звуку і багатьох інших додатках.

Найчастіше цифрова фільтрація застосовується для обробки сигналу в системах, що оперують дискретними даними.

Цифрові фільтри задаються такими характеристиками як порядок, частота зрізу, гарантоване затухання, нерівномірність.

При проектуванні фільтру стоїть задача забезпечити гарантоване затухання сигналу на частотах зрізу, і максимально прибрати нерівномірність в полосі пропускання. Забезпечити це можливо, знайшовши правильний набір коефіцієнтів передавальної функції. Тобто стоїть проблема оптимізації коефіцієнтів передавальної функції цифрового фільтру із частотною вибіркою.

На графіку функції затухання існує багато екстремумів на відносно малій області визначення. Крім того вигляд передавальної функції цифрового фільтра з частотною вибіркою є відносно складним.

Класичним методам оптимізації важко працювати в таких умовах, хоч і існують множина модифікацій. Наприклад, придумали стрибки по області визначення, аби вийти з локального екстремуму.

Сукупність цих факторів спонукала розглянути не класичні методи оптимізації, а дещо інше. В роботі розглядаються еволюційні алгоритми, а саме – генетичний для пошуку глобального максимуму.

Такий алгоритм не лякає ні кількість екстремумів, ні складність функції. Він пристосований до роботи на глобальній області визначення з багатьма екстремумами. І що важливо, що «фітнес-функція» може бути значно простіша ніж функція, з якою йде робота.

Розроблене програмне забезпечення приймає на вхід наведені вище характеристики ЦФ (порядок, частоти зрізу, гарантоване затухання, нерівномірність). Використовуючи генетичний алгоритм знаходяться коефіцієнти передавальної функції. Для зручності і продуктивності роботи буде вестися сховище даних.

1 ЗАДАЧА РОЗРОБКИ СИСТЕМИ ОПТИМІЗАЦІЇ КОЕФІЦІЄНТІВ ПЕРЕДАВАЛЬНОЇ ФУНКЦІЇ ЦИФРОВОГО ФІЛЬТРА ЗА ДОПОМОГОЮ ГЕНЕТИЧНОГО АЛГОРИТМУ

Метою розробки є створення продукту, що дозволить проектувати цифровий фільтр з оптимальними характеристиками передавальної функції. Це буде зроблено за рахунок оптимізації її коефіцієнтів за допомогою генетичного алгоритму.

Важливо зробити зручний користувацький інтерфейс, де буде чітко зрозуміло де і який параметр знаходиться, показати графіки, на яких буде зрозуміло на якому етапі проходять розрахунки (оскільки генетичний алгоритм в реальному часі шукає відповідь), і забезпечити зберігання результатів у деякому сховищі.

Задачі які мають вирішені :

- аналіз літератури стосовно проектування цифрових фільтрів з оптимальними параметрами і літератури стосовно методів глобальної оптимізації багатовимірних функцій;

- адаптувати і реалізувати генетичний алгоритм для оптимізації коефіцієнтів передавальної функції цифрового фільтра в полосі пропускання і полосах фільтрування;

- розробити програмне забезпечення для проектування цифрових фільтрів на основі частотної вибірки з оптимізацією коефіцієнтів передавальної функції;

- розробити сховище даних спроектованих фільтрів для збереження оптимальних коефіцієнтів передавальної функції для заданих параметрів цифрового фільтру.

2 АНАЛІЗ ПРОБЛЕМИ ОПТИМІЗАЦІЇ КОЕФІЦІЄНТІВ ПЕРЕДАВАЛЬНОЇ ФУНКЦІЇ ЦИФРОВОГО ФІЛЬТРА ЗА ДОПОМОГОЮ ГЕНЕТИЧНОГО АЛГОРИТМУ

Було проаналізовано різноманітну літературу, наукові статті за темою цифрових фільтрів і генетичного алгоритму.

2.1 Цифрові фільтри

Цифровий фільтр — поняття в електроніці, будь-який фільтр, що обробляє цифровий сигнал з метою відокремлення та/або придушення певних частотних складових цього сигналу. На відміну від цифрового, аналоговий фільтр має справу з аналоговим сигналом, його властивості недискретні, відповідно, передаточна функція залежить від внутрішніх властивостей його складових елементів[1].

Сьогодні цифрові фільтри застосовуються практично всюди, де потрібна обробка сигналів, зокрема у спектральному аналізі, обробці зображень, обробці відео, обробці мови та звуку і багатьох інших додатках[6].

Перевагами цифрових фільтрів перед аналоговими є:

- висока точність (точність аналогових фільтрів обмежена допусками на елементи);
- на відміну від аналогового фільтру передавальна функція не залежить від дрейфу характеристик елементів;
- гнучкість налаштування, легкість зміни;
- компактність — аналоговий фільтр на дуже низьку частоту (долі герца, наприклад) вимагав би надзвичайно громіздких конденсаторів або індуктивностей.

Фільтр зі скінченною імпульсною характеристикою (нерекурсивний фільтр, СІХ-фільтр) — один з видів електронних фільтрів, характерною особливістю якого є обмеженість по часу його імпульсної характеристики (з якогось моменту часу вона стає точно рівною нулеві). Такий фільтр називають ще нерекурсивним через

відсутність зворотного зв'язку. Знаменник передавальної функції такого фільтру — певна константа[7].

Розрізняють два види реалізації цифрового фільтру: апаратний та програмний. Апаратні цифрові фільтри реалізуються на елементах інтегральних схем, тоді як програмні реалізуються за допомогою програм, виконуваних процесором або мікроконтролером. Перевагою програмних перед апаратними є легкість втілення, а також налаштувань та змін, а також те, що у собівартість такого фільтру входить тільки праця програміста. Недолік — низька швидкість, що залежить від швидкодії процесора, а також важкість написання цифрових фільтрів високого порядку.

Що таке виділення «корисного» сигналу? Якщо відрізнити сигнал за частотою, то можна сказати, що частоти від точки $X1$ до $X2$ пропускаємо, реагуємо на них, обробляємо, а всі інші частоти — нижчі за $X1$, і вищі за $X2$ — ігноруємо. Точки $X1$ та $X2$ ще називають частотами зрізу.

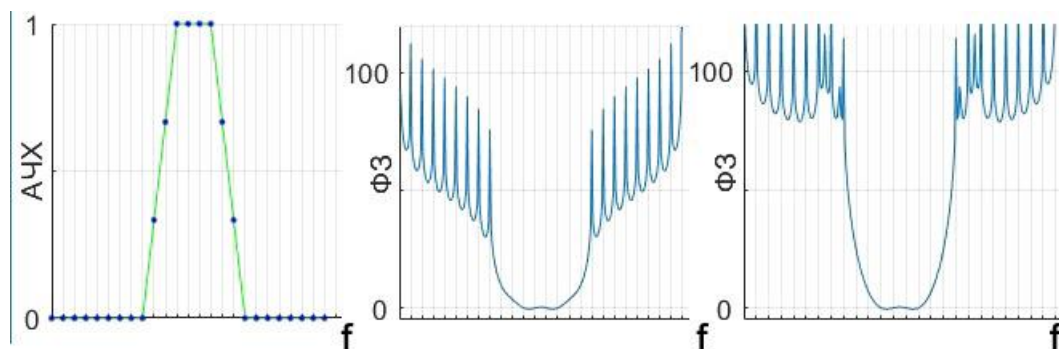


Рис.2.1 — а – АЧХ фільтра, б, в – графіки функції затухання сигналу

На рисунку 2.1-а показано, що деяка смуга частот (f) пропускається. А на рисунку 2.1-б показано за рахунок чого. Показано графік затухання. І в ідеальному фільтрі локальні мінімуми в правій та лівій полосі затримування приймають якомога найбільше середнє значення з мінімальною дисперсією.

Проте це ідеальний фільтр. А в реальному житті графік виглядає з більшими нерівностями. Це можна побачити на 2.1-в. І проектування фільтра полягає в тому, або згладити дані нерівності.

При проектуванні фільтру стоїть задача забезпечити гарантоване затухання сигналу на частотах зрізу, і максимально прибрати нерівномірність в полосі пропускання[2]. Забезпечити це можливо, знайшовши правильний набір коефіцієнтів передавальної функції. Тобто стоїть проблема оптимізації коефіцієнтів передавальної функції цифрового фільтру із частотною вибіркою.

2.2 Види цифрових фільтрів

Цифрові фільтри розділені на два великих класи: фільтри з нескінченною імпульсною характеристикою (НІХ-фільтри) і фільтри з скінченною імпульсною характеристикою (СІХ-фільтри). Фільтр кожного типу (в стандартній формі) можна представити через коефіцієнти його імпульсної характеристики $H(k)$. Вхідний і вихідний сигнали фільтра пов'язані через операцію згортки.

Фактори, що впливають на вибір з альтернатив, доступних розробнику цифрового фільтра на кожному етапі процесу проектування, значно залежать від того, до якого класу належить. Вибір між СІХ- і НІХ-фільтрами залежить від відносних переваг обох типів.

— СІХ-фільтри можуть мати строго лінійну фазову характеристику. Отже, фільтр не вводить фазового спотворення в сигнал, що важливо в багатьох сферах, наприклад, передачі даних, біомедицині, цифровій аудіо-обробці або обробці зображень. Фазова характеристика НІХ-фільтрів нелінійна, особливо на краях смуг.

— СІХ-фільтри реалізовані нерекурсивно, тобто вони завжди стійкі. Гарантувати стійкість НІХ-фільтрів вдається не завжди.

— Для реалізації фільтрів використовується обмежене число бітів. Практичні наслідки цього (наприклад, шум округлення і помилки квантування) значно менш істотні для СІХ-фільтрів, ніж для НІХ-фільтрів.

— Щоб отримати кінцеву імпульсну характеристику за допомогою фільтрів з різкими зрізами характеристики, буде потрібно більше коефіцієнтів, ніж для отримання нескінченної імпульсної характеристики. Отже, для реалізації

запропонованої специфікації амплітудної характеристики з СІХ необхідно більше обчислювальної потужності і пам'яті, ніж для реалізації її з НІХ. Втім, ефективність СІХ-реалізацій можна значно підвищити, зігравши на обчислювальній швидкості БПФ і обробці при декількох швидкостях.

— Аналогові фільтри легко перетворити в еквівалентні цифрові НІХ-фільтри, що задовольняють подібним специфікаціям. Для отримання СІХ-фільтрів таке перетворення неможливо, оскільки для них не існує аналогових прототипів. Втім, отримувати довільні частотні характеристики на СІХ-фільтрах легше.

— Взагалі, синтез СІХ-фільтрів алгебраїчно складніший якщо не використовувати комп'ютерну підтримку розробки.

— НІХ-фільтри рекурентності. Це означає, що, пропустивши через фільтр один і той же сигнал, але з «зворотним ходом часу», ми отримаємо, взагалі кажучи, різні результати. Якщо для мови тимчасова анізотропія природна, то, наприклад, для зображень вже немає, тому НІХ-фільтри мають ряд обмежень щодо застосування.

З огляду на наведені міркування, загальний принцип вибору між СІХ і НІХ можна сформулювати наступним чином.

— Використовувати НІХ, якщо єдиними важливими вимогами є характеристика з різкими зрізами і висока пропускну здатність, оскільки НІХ-фільтри (особливо ті, в яких використані еліптичні характеристики) зажадають означення меншої кількості коефіцієнтів, ніж СІХ-фільтри.

— Використовувати СІХ, якщо число коефіцієнтів фільтрів не дуже велика і, зокрема, якщо потрібно, щоб фазове спотворення відсутнє або було малим. Крім того, можна додати, що архітектури новітніх процесорів ЦГЗ пристосовані до СІХ-фільтрації, мало того, деякі з них спеціально розроблені для СІХ-фільтрів.

2.3 Методи проектування цифрових фільтрів

Розробка цифрового фільтру проходить в п'ять етапів.

1. Специфікація вимог до фільтру.

2. Обчислення відповідних коефіцієнтів фільтра.
3. Подання фільтра придатною структурою.
4. Аналіз впливу кінцевої розрядності на продуктивність фільтра.
5. Реалізація фільтра на програмному та / або апаратному рівні.

Названі п'ять етапів не завжди незалежні; крім того, вони не завжди розташовуються в зазначеному порядку. Фактично існують методи, які дозволяють об'єднати другий етап і деякі аспекти третього і четвертого. Щоб отримати ефективний фільтр, іноді доводиться проводити даний процес в кілька ітерацій, особливо, якщо специфікації не є абсолютно певними (як зазвичай і буває), або ж розробник бажає досліджувати альтернативні структури.

Специфікація вимог включає специфікації.

— характеристик сигналів (тип джерела і одержувача сигналу, інтерфейс введення-виведення, швидкість передачі даних і ширина смуги, найвища частота, що представляє практичний інтерес);

— характеристик фільтра (бажана амплітудна і / або фазова характеристика і то, наскільки дані вимоги суворі, швидкість роботи і режими фільтрації (реальне або модельне час));

— принципу реалізації (наприклад, як комп'ютерної програми на мові високого рівня або як системи ЦОС на базі процесора, тут же виконується вибір процесора сигналів);

— інших вимог до структури (наприклад, вартість фільтра).

Розробник може не мати достатньо інформації щоб повністю визначити фільтр на початкових етапах, але для спрощення процесу розробки слід сформулювати максимальну кількість вимог. Хоча перераховані вище вимоги уточнюються в залежності від конкретного завдання, деяким аспектам п. 2 варто приділити особливу увагу. Характеристики цифрових фільтрів часто задаються в частотній області. Для частотно-виборчих фільтрів, таких як фільтри нижніх частот і смугові фільтри, специфікації часто формулюються у вигляді схем допусків. Приклад такої схеми для фільтра нижніх частот наведено на рисунку 2.2.

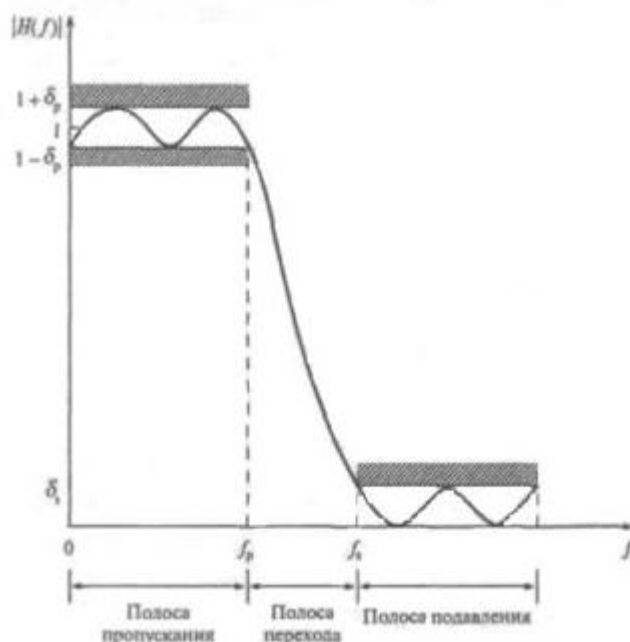


Рисунок 2.2 — Приклад схеми фільтра нижніх частот

На етапі підрахунку коефіцієнтів вибирається один з методів апроксимації та обчислюються значення коефіцієнтів $h(k)$ (для СІХ-фільтра) або a_k і b_k (для НІХ-фільтра), при яких задовольняються умови, принципи визначення яких представлені були вище. Метод обчислення коефіцієнтів фільтра залежить від того, до якого класу належить фільтр – СІХ або НІХ.

Обчислення коефіцієнтів НІХ-фільтра традиційно ґрунтується на перетворенні характеристик відомих аналогових фільтрів в характеристики еквівалентних цифрових. При цьому використовуються два основні підходи: метод інваріантного перетворення імпульсної характеристики і метод білінійної перетворення. При використанні методу інваріантного перетворення імпульсної характеристики після оцифровки аналогового фільтра зберігається імпульсна характеристика вихідного аналогового фільтра, але не зберігається амплітудно-частотна характеристика. Внаслідок внутрішнього накладення даний метод не підходить для фільтрів верхніх частот або режекторного фільтрів. Білінійний метод, з іншого боку, забезпечує досить ефективні фільтри і добре підходить для обчислення коефіцієнтів частотно-виборчих фільтрів. В результаті можна створювати цифрові фільтри з відомими класичними характеристиками, такими як у фільтрах Баттерворта, Чебишева або еліптичних.

Цифрові фільтри, отримані методом білінійної перетворення, будуть, в загальному випадку, мати ту ж амплітудну характеристику, що і аналогові, але інші властивості в тимчасовій області. Метод інваріантного перетворення імпульсної характеристики хороший при моделюванні аналогових систем, але для частотно-виборчих НІХ-фільтрів краще використовувати білінійну метод.

В якості альтернативної схеми обчислення коефіцієнтів НІХ-фільтрів застосовується ще метод розміщення нулів і полюсів - простий шлях обчислення коефіцієнтів дуже простих фільтрів. У той же час, для фільтрів з хорошою амплітудною характеристикою даний метод використовувати не рекомендується, оскільки в ньому фігурує перебір положень нулів і полюсів.

Коефіцієнти СІХ-фільтрів також можна обчислити декількома різними способами. У наступних лабораторних роботах ми розглянемо три методи: вирізання (зважування), частотної вибірки і оптимальний (алгоритм Паркса-Мак-Кліллана (Parks-McClellan)). Метод зважування надає дуже простий і гнучкий спосіб обчислення коефіцієнтів СІХ-фільтра, але не дозволяє розробнику адекватно керувати параметрами фільтра. Найпривабливішою рисою методу частотної вибірки є те, що він допускає рекурсивну реалізацію СІХ-фільтрів, що може бути дуже обчислювально вигідно. У той же час, цим методом бракує гнучкості в плані управління або завдання параметрів фільтрів. В даний час в промисловості широко використовується оптимальний метод, який в більшості випадків дає необхідний СІХ-фільтр. Отже, при проектуванні таких фільтрів спочатку варто спробувати оптимальний метод, якщо тільки конкретне застосування не передбачає використання іншого методу.

Існує кілька методів обчислення коефіцієнтів фільтрів, найбільш використовуваних з них перераховані нижче.

- метод інваріантного перетворення імпульсної характеристики (НІХ);
- білінійне перетворення (НІХ);
- розміщення нулів і полюсів (НІХ);
- метод зважування (СІХ);
- частотна вибірка (СІХ);

— оптимізаційні методи (СІХ).

На вибір методу, що найбільш підходить для конкретного завдання, впливають кілька факторів, зокрема критичні вимоги в специфікаціях. Взагалі, основним є вибір між СІХ і НІХ. Якщо наявність кінцевої імпульсної характеристики істотно, то варто використовувати оптимальний метод, якщо ж бажана нескінченна імпульсна характеристика, то в більшості випадків буде достатньо білінійного методу.

2.4 Актуальність цифрової фільтрації

Цифрова фільтрація – важлива частина сучасного життя. Взяти, наприклад, мобільний телефон, яким зараз усі користуються. Цей пристрій може програвати аудіо і відео, обробляти фото, передавати і приймати по комп’ютерній мережі, оцінювати власне місце перебування в просторі, сканувати відбиток пальця або обличчя користувача. І все це реальність саме завдяки цифровій обробці сигналів.

Прикладів таких в нашому житті багато. Сучасні медичні пристрої, розумні транспортні засоби обробляють сигнали з багатьох датчиків і самостійно оцінюють дорожню ситуацію. Радарні станції, що контролюють посадку літаків на аеродромах, телекомунікації і системи глобального позиціонування – все це опирається на алгоритми цифрової фільтрації.

Іншими словами, цифрова фільтрація застосовується у всьому, що знаходиться навколо нас. І від неї залежить наш комфорт і безпека.

2.5 Порівняння існуючих рішень

Один з конкурентів поточної програми – це t-filter. Веб-версію цієї програми можна побачити на рисунку 2.3 :

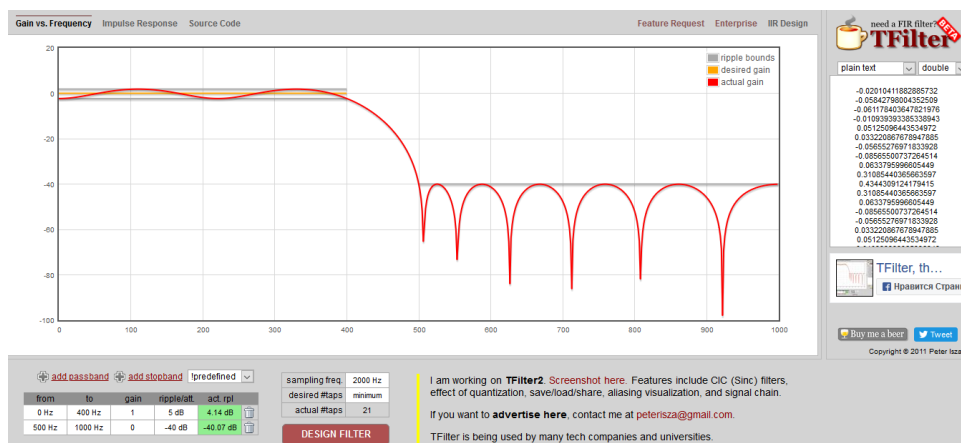
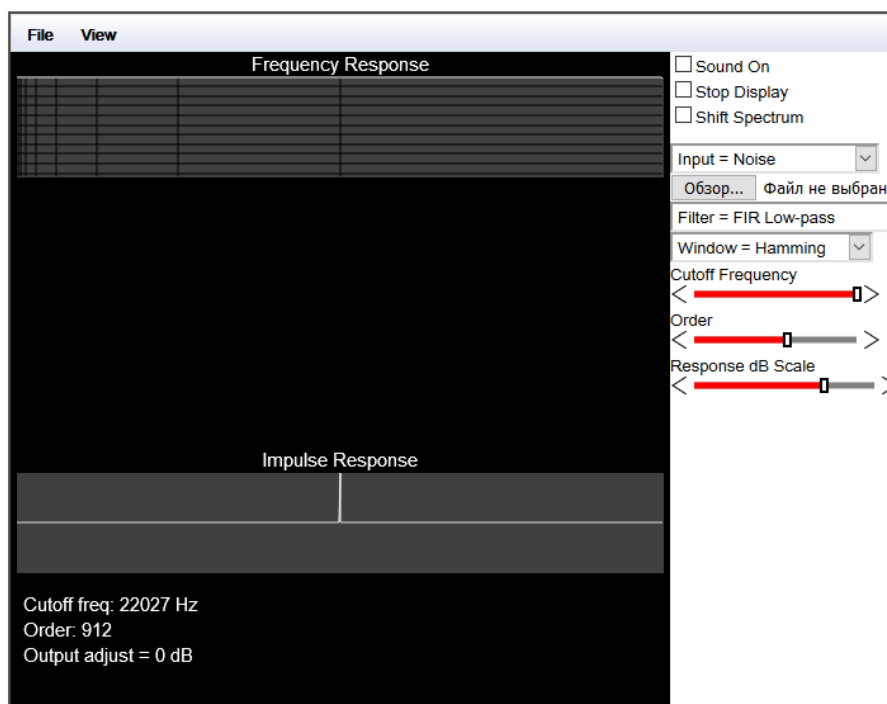


Рисунок 2.3 — Програма T-Filter

У програмі є множина налаштувань. Щодо початкових параметрів перед обчисленнями. А також для виокремлення потрібних результатів після отримання результатів. Можна додавати смуги (якщо смуговий фільтр). Можна вибрати режим «низькочастотний фільтр», чи «високочастотний». Існує два вікна для перегляду результатів: частотне порівняння та імпульсна характеристика.

Інший аналог — це Falstad.



This applet is a demonstration of digital filters. Click the "Sound On" checkbox to turn the sound on.

Рисунок 2.4 — Програма Falstad

Крім того, є ще десктоп-програми для проектування фільтрів. Найвідоміший в сфері освіти – це Matlab.

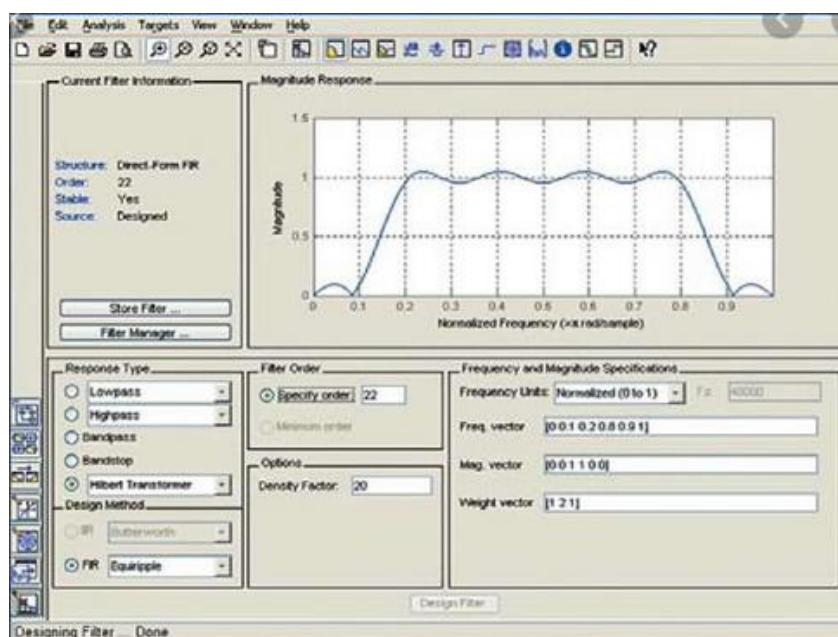


Рисунок 2.5 — Приклад проектування цифрового фільтру в Matlab

2.6 Сучасні підходи до побудови методів глобальної оптимізації

Оптимізація в широкому тексті слова знайде застосування в науці, техніці та в будь-якій іншій області людської діяльності. Це може бути завданням проектування, задачі розподілу обмежених ресурсів, завданням розповсюдження траєкторії польових ракет і т.д. Підтверджені задачі часто зустрічаються в ряді прикладних областей при моделюванні реальних процесів: теорія управління, моделювання фізичних ядер, аналіз даних і інших областей, слів, будь-де, де необхідно отримати більш ефективний функціонал на ряді окремих обмежень.

Розглянемо задачу глобальної оптимізації :

$$F(x) \rightarrow \min_{x \in X}, \quad (2.1)$$

Де багатоекстримальна цільова функція $F(x)$ являється випуклою і для різних алгоритмів має різні властивості гладкості, $X \subset R_k$ – допустима множина задачі, R_k – k -мірний Евклідовий простір.

Множини :

$$X^{opt} = Arg \min_{x \in X} F(x), \quad (2.2)$$

$$X^{lopt} = \{x \in X \mid \exists \varepsilon > 0 : F(x) \leq F(x'), \forall x' \in B_\varepsilon(x) \cap X\}$$

Це множини оптимальних рішень і множини локально-оптимальних рішень відповідно, і

$$B_\varepsilon(x) = \{x' \in R_k \mid |x' - x| < \varepsilon\}. \quad (2.3)$$

Таким чином задача глобальної оптимізації є в тому, аби знайти точки $x \in X^{opt}$, причому в тому випадку, коли $X^{lopt} \neq X^{opt}$.

В силу широти класу багатоекстремальних функцій завдання глобальної оптимізації в загальному випадку є нерозв'язним, тобто не можна гарантувати, що рішення задачі буде отримано за кінцеве число кроків. Навіть деякі з вирішуваних завдань можуть потрапити в клас нерозв'язних, тому що число кроків, необхідних для отримання рішення, може бути надмірно великим. Для того щоб вирішити завдання, крім безперервності цільової функції, необхідні деякі додаткові умови її гладкості. Таким чином, специфіка завдання глобальної оптимізації полягає в багатоекстремальності цільової функції і нерозв'язності в загальному випадку.

Існує два напрямки в роботі з нерозв'язними завданнями. Перший спосіб - коли формулюються апріорі відомі умови, накладені на цільову функцію і допустиму множину завдання, які призводять вихідну задачу до такого виду що можна розв'язати або, принаймні, дають можливість з упевненістю говорити про те, що точне рішення буде знайдено. Це звужує розглянутий клас функцій. Другий підхід дозволяє розглядати ширший клас цільової функції: відмовляючись від вимоги

розв'язання задачі, необхідно отримати оцінку глобального рішення. При такому підході бажано мати також деякий критерій прийнятності отриманої оцінки.

Для отримання точного рішення «істотно» безумовних завдань глобальної оптимізації було запропоновано багато методів. Точні методи або покладаються на апріорну інформацію про те, наскільки швидко змінюється функція (тобто необхідна інформація про константи Лейбніца функції), або вимагають аналітично сформулювати функцію мети (наприклад, в методі інтервалів). Статистичні методи, як правило, використовують техніку розбиття, щоб розділити область пошуку, але такі методи також користуються апріорною інформацією або деякими припущеннями про те, як функція мети може бути промодельована. Основне припущення полягає в тому, що кожна конкретна цільова функція належить класу функцій, які промодельовані конкретної стохастичною функцією. Інформація з попередньої вибірки цільової функції може бути використана для оцінки параметрів стохастичною функції, і ця вдосконалена модель згодом може бути використана для зміщення при відборі точок в досліджуваній області.

Розвиток методів глобальної оптимізації стимулюється не тільки актуальністю або складністю цих завдань, а й розвитком електронно-обчислювальних засобів. В даний час паралельні і векторні суперкомп'ютери розглядаються як один з основних інструментів для проведення досліджень в різних наукових і прикладних дисциплінах. Раніше паралелізм розглядався як рідкісна і екзотична область обчислень, цікава, але не призначена для «простих» користувачів. Аналіз тенденцій розвитку прикладних програм, архітектур обчислювальних систем, мереж показує, що ця точка зору вже невірна. Комп'ютери що виконують паралельні обчислення вже не рідкість і останні дослідження в оптимізації беруть до уваги архітектурні особливості сучасних комп'ютерів, на яких ці алгоритми передбачається реалізувати. Проблемам паралельних обчислень в задачах глобальної оптимізації, а також різним підходам до розпаралелювання алгоритмів глобальної оптимізації в даний час надається велике значення [8], [9].

Одночасно з виникаючими завданнями і потребами в їх вирішенні ведуться роботи по розробці нових і вдосконаленню існуючих спеціальних інструментів, що

дозволяють вирішувати ці проблеми, використовуючи весь накопичений досвід і сучасні обчислювальні засоби. Проблемам глобальної оптимізації, розробці нових методів і іншим важливим дослідженням в глобальній оптимізації присвячено чимало робіт [10], [11].

2.7 Властивості методів глобальної оптимізації

Процедуру (або алгоритм) рішення задачі оптимізації можна представити у вигляді ітеративного процесу, який породжує послідовність точок відповідно із запропонованим набором правил, що включає критерій закінчення рахунку. Зазвичай, глобальне рішення задачі оптимізації пропонується знайти, перебравши всі її локальні рішення. Таке завдання, як правило, виявляється трудомістким. Інший підхід - перебрати частину локальних рішень і показати, що ті що залишилися локальні мінімуми не впливають на точність рішення. Таким чином, ідея всіх методів глобальної оптимізації - оцінити значення цільової функції $F(\cdot)$ на деякій множині точок x_1, \dots, x_N з допустимої множини точок X , і різних методів заключається в способах вибору цих точок.

Оскільки ми не знаємо, де саме в X можна знайти глобальні мінімуми, то необхідно застосувати деяку стратегію, щоб «розкидати» ці точки по множині X . Будь-яку таку стратегію називають глобальною технікою (global technique). Далі, цілком можливо, що в околиці обраної точки існує краще значення цільової функції $F(\cdot)$. Для отримання кращого значення функції застосовують алгоритм локального спуску. Метод, який здійснює такий локальний спуск, називають локальною технікою (local technique). Майже всі методи глобальної оптимізації використовують методи локальної оптимізації, принаймні, для того щоб поліпшити вже знайдену оцінку глобального рішення, тому незалежно від застосовуваної глобальної техніки, використання локальної технікою є важливою частиною будь-якого методу глобальної оптимізації.

Іншою важливою властивістю алгоритму оптимізації є збіжність генерованої ним послідовності точок до глобального оптимального рішення. Однак в більшості випадків виходять менш сприятливі результати: неопуклість функцій, велика розмірність завдання або інші труднощі змушують зупиняти алгоритм, якщо отримана точка, що належить деякому множині наближених рішень. Іншими словами, для будь-якого чисельного алгоритму необхідні умови зупинки. Це ключовий момент, тому що без будь-якої додаткової інформації або припущень про завдання неможливо зробити висновки про точність рішення, отриманого за деякий фіксований число кроків алгоритму. Таким чином, для чіткого визначення умов зупинки алгоритму необхідна додаткова інформація або припущення, або ж умови зупинки повинні бути наближеними. Це підтверджує висловлений раніше факт, що завдання глобальної оптимізації в загальному випадку нерозв'язне, і ми повинні бути готові прийняти отримане за допомогою чисельного методу наближення за вирішення завдання. Таким чином, отримати чисельний розв'язок задачі глобальної якось точку, де оптимізації означає отримати

$$x \in X^{opt}(\varepsilon) = \{x \in X \mid F(x) \leq F_{opt} + \varepsilon\} \text{ де } F_{opt} = \min_{x \in X} F(x), \quad (2.4)$$

а величина ε невідома.

Будь-чисельний метод має свої переваги і недоліки, в зв'язку з чим виникає питання про порівняння різних методів. Існує ряд факторів, які слід враховувати при оцінці ефективності алгоритмів і їх порівнянні. Так, універсальність алгоритму визначається тим класом задач, для вирішення яких він призначений, а також рамками вимог, що пред'являються алгоритмом до завдань даного класу. Іншими важливими характеристиками алгоритму є його надійність (або стійкість), точність, чутливість до параметрів і вихідних даних, витрати на попередню обробку і обчислення.

Крім того, з появою паралельних обчислювальних систем виникає інше цікаве питання: як порівнювати паралельний і послідовний алгоритми. Не викликає сумніву

факт, що порівнювати паралельні алгоритми необхідно по-своєму [8]. Для оцінки ефективності паралельних алгоритмів використовують різні підходи, найбільш поширеним з яких є показник прискорення (speedup). Прискорення (speedup), що отримується при роботі алгоритму на p процесорах - це відношення часу роботи алгоритму на одному процесорі до часу роботи того ж алгоритму на p процесорах. Лінійне прискорення (linear speedup) спостерігається, коли паралельний алгоритм на p процесорах працює в p разів швидше, ніж на одному процесорі. Сублінійне прискорення (sub-linear speedup) досягається, коли поліпшення в швидкості рахунку менше p . Суперлінійне прискорення (super-linear speedup) досягається, коли поліпшення в швидкості рахунку більше p . Закон Амдала [12] дозволяє обчислити верхню межу прискорення, яке можна очікувати від паралельної реалізації алгоритму.

2.8 Методи глобальної оптимізації

На жаль, для вирішення завдання глобальної оптимізації не існує універсального по ефективності алгоритму. Тому при розробці специфічних методів глобальної оптимізації в першу чергу враховують властивості цільової функції $F(\cdot)$ допустимої множини X розглянутого класу задач, для яких розробляється метод.

Всі відомі методи глобальної оптимізації можна розділити на дві категорії: детерміновані [13] і стохастичні [14]. Детерміновані методи набувають глобального рішення за допомогою вичерпного пошуку на всій допустимій множині. Тому більшість детермінованих методів втрачають ефективність і надійність зі зростанням розмірності задачі. Більш того, щоб гарантувати успіх, такі методи вимагають виконання додаткових припущень, накладених на цільову функцію. Детерміновані методи не використовують стохастику.

Стохастичні алгоритми дозволяють піти від проблем детермінованих алгоритмів. Тут стохастичний підхід присутній не тільки в розробці і аналізі алгоритму, але і використовується в рішенні базових проблем, наприклад, при визначенні умови зупинки. Більшість стохастичних методів оцінюють значення

функції мети в випадкових точках допустимої множини з подальшою обробкою вибірки. Як наслідок, стохастичні методи не гарантують успіх.

Нижче наведені деякі приклади сучасних методів глобальної оптимізації. Опис методів оптимізації також можна знайти в [13], [15].

Історично першим методом глобальної оптимізації є метод Монте-Карло, на базі якого був створений метод мультистарту (multistart method) [14]. У методі мультистарту з множини X випадково або детерміновано вибирається деяка підмножина з N точок. Послідовно з кожної точки запускається алгоритм локального спуску, і з отриманої множини локальних рішень вибирається найкраще. У чистому вигляді метод мультистарту не є ефективним, тому що одне і те ж локальне рішення може бути знайдено не один раз. Мультистарт - це узагальнений підхід: більшість ефективних методів глобальної оптимізації засноване на ідеї методу мультистарту - запуску стандартних локальних алгоритмів з множини точок, рівномірно розподілених на множині X . Таким чином, метод мультистарту можна назвати прототипом таких методів.

Методи угруповань (clustering methods) [15] є однією з модифікацій методу мультистарту. Тут зроблено спробу усунення головного недоліку мультистарту шляхом ретельного відбору точок, з яких запускається локальний пошук. Розглядається деяка вибірка точок, наприклад, рівномірно розподілених на X . Потім, поки не виконається деяка умова зупинки, послідовно виконуються наступні три кроки:

- 1) з кожної точки запускається алгоритм локального спуску, в результаті буде отриманий набір локальних рішень;
- 2) використовуючи спеціальну техніку угруповання, визначаються групи точок;
- 3) в якості нової вибірки точок розглядається кожна m -а точка з групи, і здійснюється перехід до першого кроку.

Таким чином, рішення знаходяться за допомогою локальних алгоритмів спуску з найкращих точок кожної групи.

Існують різні модифікації оригінального методу угруповань, розроблені багатьма авторами. Одна з таких модифікацій, де найкращий елемент групи визначається безпосередньо, без застосування техніки угруповання, носить назву топографічного методу (topographical method) [16]. Ідея топографічного методу складається з трьох концептуальних кроків. На першому кроці розглядається довільна вибірка точок, рівномірно розподілених на допустимій множині X . На другому кроці необхідно визначити деякій множині S допустимих точок. Це робиться за допомогою побудови так званої топографії. Топографія є спрямований граф з точок вибірки першого кроку, в вершинах якого зберігається інформація про цільової функції в даній точці. Кожна вершина графа з'єднується спрямованими дугами з сусідніми вершинами, в яких значення цільової функції краще. Мінімуми, тобто вершини, з яких не виходить жодної дуги, в графі утворюють множину S . Третій крок визначає локальні мінімуми, представлені множиною S : з кожної точки цієї множини запускається локальний алгоритм. Найкраща отримана точка буде рішенням вихідної задачі. Використовувана топографія дає можливість реалізації топографічного алгоритму на машинах з паралельною архітектурою. Результати роботи послідовних і паралельних версій топографічного методу представлені в [16].

Методи поділу навпіл (bisection methods) і методи інтервалів (interval methods) [17], [18], [19], [20], [21] гарантують, що рішення буде отримано із заданою точністю. Ці методи в глобальній оптимізації також носять назву методів покриттів (covering methods) або методів гілок і меж (branch-and-bound methods). Ціна пропонованої методами гарантії - деяка а priori відома інформація про функції мети. Так, для методу розподілу навпіл необхідне знання константи Ліпшиця функції мети, а для методу інтервалів функція мети повинна бути двічі безперервно диференціюється, і перша і друга похідні повинні мати кінцеве число нулів. Більшість методів інтервалів використовують стратегію гілок і меж (branch-and-bound strategy) [20]. Такі алгоритми поділяють область пошуку на набір багатовимірних кубиків, на яких нижня межа функції мети обчислюється за допомогою інтервального техніки. Використовуючи інтервальну арифметику на кожному кроці, отримуємо набір послідовно зменшуються інтервалів, який містить глобальне рішення вихідної задачі.

Алгоритм зупиняється, коли розмір інтервалів досягає заздалегідь заданого значення [22]. Методи інтервалів вимагають, щоб функція мети була задана явно, тому що це вираз використовується інтервального технікою для обчислення меж. Методи, які використовують техніку гілок і меж, в силу деревовидної структури є об'єктом досліджень в контексті паралельних обчислень. Деякі розробки паралельних алгоритмів гілок і меж наводяться в [23].

Кожен крок методу тунелів (tunneling method) [24] складається з двох фаз: фаза мінімізації покращує значення поточного рекорду; фаза тунелювання знаходить точку з допустимої множини, відмінну від останньої, де знайдено значення мінімуму. Отримана на фазі тунелювання точка розглядається як стартова для чергового циклу. Головний недолік цього алгоритму - необхідність вирішення складних нелінійних диференціальних рівнянь.

Основна ідея методу моделювання випалу (simulated annealing) виходить з фізики процесу замерзання рідин або рекристалізації металів в процесі випалу. Цільова функція тут є аналогом рівноваги термодинамічної системи і видозмінюється шляхом додавання випадкових величин (умов температурного режиму) [23]. Процес повторюється достатню кількість разів для кожної температури, після чого температура знижується і весь процес відбувається знову до стану повної заморозки. Уникнення потрапляння в незначні локальні мінімуми (замерзання) залежить від «схеми випалу», вибору початкової температури, кількості ітерацій для кожної температури і наскільки зменшується температура на кожному кроці процесу «охолодження».

Еволюційні алгоритми (evolutionary algorithms) [24] є пошуковими методами, основна ідея яких запозичена з біологічного процесу природного відбору і процесу виживання. Такі алгоритми відрізняються від традиційних методів оптимізації тим, що пошук проводиться з «популяції» рішень, а не з однієї точки. Кожна ітерація методу проводить «природний відбір», який відсіває невідповідні рішення. Рішення з високою придатністю («біологічною реакцією на природний відбір») «схрещуються» з іншими рішеннями шляхом обміну частин одних рішень на інші. Рішення можуть «мутувати» через невеликі заміни одного елемента рішення. Схрещування і мутації

генерують нові рішення, які «генетично» налаштовані на області допустимої множини, для яких уже було виявлено гарне рішення. Існує кілька різних типів еволюційних пошукових алгоритмів: алгоритми генетичного програмування (genetic programming); алгоритми еволюційного програмування (evolutionary programming); алгоритми еволюційних стратегій (evolutionary strategies); генетичні алгоритми (genetic algorithms). Еволюційні алгоритми мають слабку збіжність до глобального вирішення, але в той же час, добре обробляють сильно зашумлені функції з великим числом незначних локальних рішень, не "прилипають" до локальних екстремумів і здатні отримати глобальне рішення.

Алгоритм контрольованого випадкового пошуку (controlled random search algorithm) [25] є методом прямого пошуку. Алгоритм починає роботу на множині S початкових точок, рівномірно розподілених на допустимій множині завдання. Багаторазово утворюються нові відчувачі точки, які заміщають підмножина W гірших точок з множини S в разі, якщо вони краще. Алгоритм зупиняється, коли всі отримані значення функції досить близькі. Існує багато модифікацій оригінального методу: відмінності полягають в способі вибору випробовуваних точок і використанні локальної мінімізації. Останні модифікації для генерації випробовуваних точок використовують квадратичну апроксимацію замість симплексного методу [26]. Хоча алгоритми прямого пошуку є стійкими до перешкод різного роду, вони вимагають набагато більше обчислень функції, ніж алгоритми, що використовують градієнт. Методи прямого пошуку добре підходять для паралельних обчислень [27].

Траєкторні методи (trajectory methods) [27], [28], [29], [30] і, зокрема, метод продовження (continuation method) займають важливе місце в глобальній оптимізації. Траєкторні методи на допустимій множині вихідної задачі будують множину кривих (траєкторій). Основна ідея траєкторних методів полягає в тому, що всі рішення вихідної задачі а пріорі лежать на цих кривих. У багатьох випадках ці траєкторії є рішенням систем звичайних диференціальних рівнянь першого або другого порядку.

2.9 Генетичний алгоритм

Генетичний алгоритм — це еволюційний алгоритм пошуку, що використовується для вирішення задач оптимізації і моделювання шляхом послідовного підбору, комбінування і варіації шуканих параметрів з використанням механізмів, що нагадують біологічну еволюцію.

Особливістю генетичного алгоритму є акцент на використання оператора «схрещення», який виконує операцію рекомбінацію рішень-кандидатів, роль якої аналогічна ролі схрещення в живій природі. «Батьком-засновником» генетичних алгоритмів вважається Джон Голланд книга якого «Адаптація в природних і штучних системах» є фундаментальною в цій сфері досліджень[3].

У генетичному алгоритмі використовуються такі біологічні поняття як популяція, хромосома, гени, схрещування, мутація. На початку кожної ітерації існує популяція хромосом. З популяцією роблять схрещування, генеруючи нащадків(в їх генах роблять мутацію для різноманіття популяції)[4]. Маючи батьків і дітей, за допомогою «фітнес-функції» відбирають найкращих особин для наступної ітерації. Приклад основних операцій можна побачити на рисунку 2.6 :

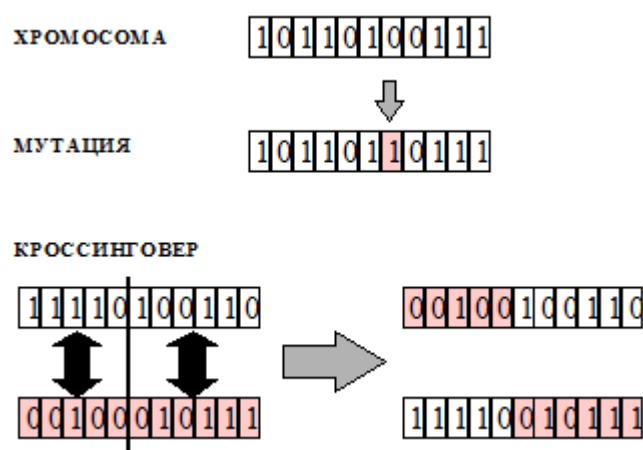


Рисунок 2.6 — Основні операції генетичного алгоритму

Такий алгоритм не лякає ні кількість екстремумів ні складність функції. Він пристосований до роботи на глобальній області визначення з багатьма екстремумами.

І що важливо, що «фітнес-функція» може бути значно простіша ніж функція, з якою йде робота[5].

Під час розробки програми, було виділено такі параметри для генетичного алгоритму як : розмір популяції – 60. Кількість ітерацій – 30. Хоча це не є максимальне значення для знаходження результату. Адже особливістю алгоритму є удача. То з першої ітерації отримаємо високий результат, то з 30-ої. Але звичайно ця удача налаштовується параметрами. Тільки у вікно налаштувань було винесено велику кількість таких різних параметрів.

2.10 Результати експериментів під час розробки

Під час розробки програми було зроблено деякі метрики. Наприклад, знайдено константний час виконання простого перебору для однієї соті (точність). І це 2 хвилини. А для однієї тисячної – 8 хвилин. Очевидно що це довго. І треба було б знайти краще рішення. І генетичний алгоритм – гарний вибір. Він зазвичай показує кращий результат за швидкістю обчислень, ніж перебір. Іноді взагалі може з першої ітерації показати гарний максимум.

Стосовно параметрів, які знаходяться, проводились дослідження, і було знайдено, що перший коефіцієнт знаходиться між 0 і 0.1, а другий – 0.4 – 0.5. Знання цього значно спрощує величину пошуку.

Висновок до розділу 2

Цифрова фільтрація є невід’ємною частиною нашого життя. І якщо колись була просто потреба у її актуальності, то з часом постало питання як же зробити оптимальну фільтрацію.

Існує декілька класифікацій цифрових фільтрів. А також багато способів їх проектування. Частотна вибірка – один з них.

Одна з проблем проектування цифрових фільтрів – це вигляд функції затухання. Вона багатоекстримальна. Тому звичайні методи оптимізації тут не ефективні. Потрібна глобальна оптимізація.

Завдання розробки нових і вдосконалення існуючих методів глобальної оптимізації залишається відкритим. Для вирішення завдання глобальної оптимізації не існує універсального по ефективності алгоритму. Тому при розробці специфічних методів глобальної оптимізації в першу чергу враховують властивості конкретного класу задач, для яких розробляється метод. Методам глобальної оптимізації присвячено багато робіт.

В даний час розробка методів оптимізації ведеться не тільки у відповідності з існуючими потребами у вирішенні конкретних завдань, але і враховується сучасний стан обчислювальної техніки. Паралельні обчислення представляють собою потужний засіб для проведення наукових експериментів.

Метод глобальної оптимізації, який використовується в роботі – генетичний алгоритм. Його вже достатньо добре вивчили і розібрали і в літературі і на практиці.

3 ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ

Структура програми складається із трьох частин. Перша – базова – призначена для проведення розрахунків. Наприклад, клас – Calculations містить усі формули, що потрібні для знаходження графіка гарантованого затухання. А це такі функції як H10, H18, H111, АЧХ. Наступна логічна частина – побудова графіків. І остання, найбільш важливіша – реалізація генетичного алгоритму.

На вхід користувач може ввести такі параметри як порядок фільтру, кількість розрядів квантування, і коефіцієнти. Останні насправді не дуже й потрібно, тому що їх програма може розрахувати. В реалізації даної програми існує 8 коефіцієнтів, а це – 4 в зоні пропускання, які завжди дорівнюють 1, а ще по 2 в транзитних зонах – це ті коефіцієнти, які потрібно знайти. Їх значення коливаються від 0 до 1.

3.1 Основи генетичного алгоритму

Основна задача програми – оптимізації. Для її реалізації використовується генетичний алгоритм. Тому розпочнімо з нього. За допомогою нього оптимізується значення двох коефіцієнтів в транзитних полосах. Два коефіцієнти являють собою таку структуру як хромосома.

Гени реалізовані через рядки – клас String, і складаються з нулів і одиниць. І це не зовсім двійкова система числення. Адже функція перетворення такого рядка в число, може знайти число в будь-якому діапазоні при будь-якій довжині рядка. Але за замовчуванням виставлено 16 символів в гені, і діапазон значень – від 0 до 1. І при цьому різних чисел від може бути 65536. Доволі велике число як для чисел від 0 до 1. Але це дозволяє отримати достатню точність. А вона необхідна, оскільки при малій кількості знаків після коми (наприклад, кількість генів 4), навіть велика кількість ітерацій не може збільшити фітнес-функцію.

Що собою представляє фітнес-функція? Це значення гарантованого затухання.

3.2 Пошук гарантованого затухання

Існує клас `GuaranteedAttenuationFinder` для знаходження гарантованого затухання. В ньому важливі 2 методи : «отримати точку гарантованого затухання» і «отримати точки мінімуму». Точки мінімуму визначаються так : перебираються всі точки, і якщо в якоїсь точки сусідні є вищими, отже це необхідна нам точка.

Маючи точки мінімуму є декілька етапів які необхідно пройти аби отримати значення гарантованого затухання.

Перший етап – відсіяти такі точки мінімуму, які не вписуються в загальну тенденцію. А вона така – зліва направо в лівій полосі затримки мінімуми спадають, а в правій полосі – зростають. На рисунку 3.1 зображено показовий графік, який відповідає даним вимогам. На ньому червоними точками позначено мінімуми. А жовта лінія – показує лінію гарантованого затухання.

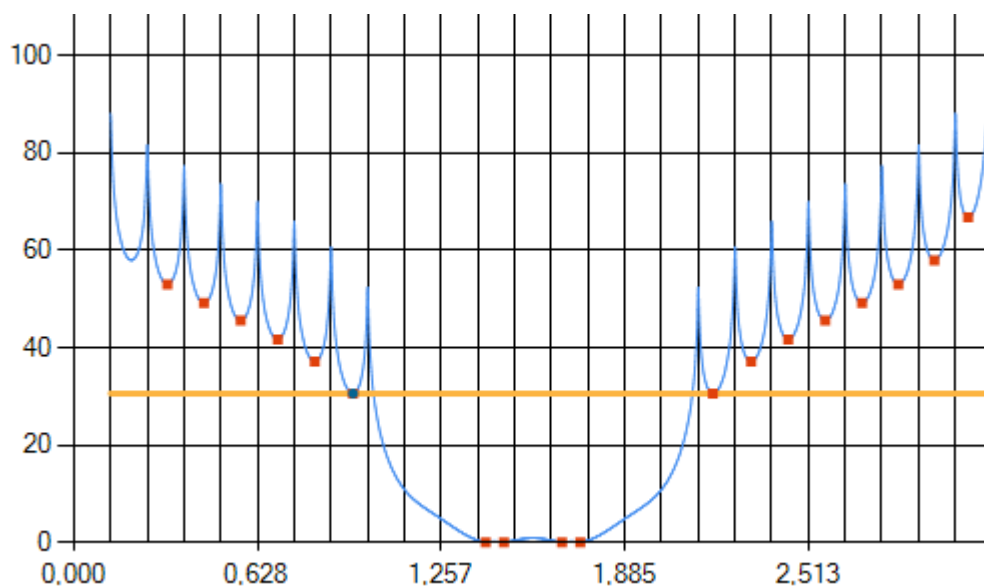


Рисунок 3.1 — Графік гарантованого затухання

Якщо не прибрати такі точки мінімуму, сусіди яких нижчі, тоді буде щось як на рисунку 3.2 :

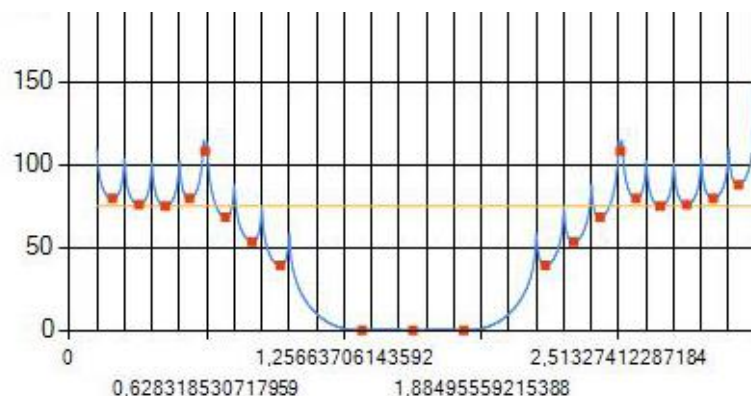


Рисунок 3.2 — Неправильно визначене гарантоване затухання

Такий результат як на рисунку 3.2 не прийнятний. Знайдено найперші локальні мінімуми вліво і вправо від полоси пропускання і проведено лінію гарантованого затухання. Але нижче цієї лінії є локальні мінімуми. Тобто лінія гарантованого затухання проводиться по найнижчому локальному мінімуму в зонах затримки.

Вважається, що оптимальне значення досягається коли, цей найменший локальний мінімум найближче до полоси пропускання, а всі інші локальні мінімуми в полосах затримки наближаються зверху до горизонтальної прямої.

Далі видаляється перший мінімум, і останній. Якщо цього не зробити, то результат може виявитись таким як на рисунку 3.3 :

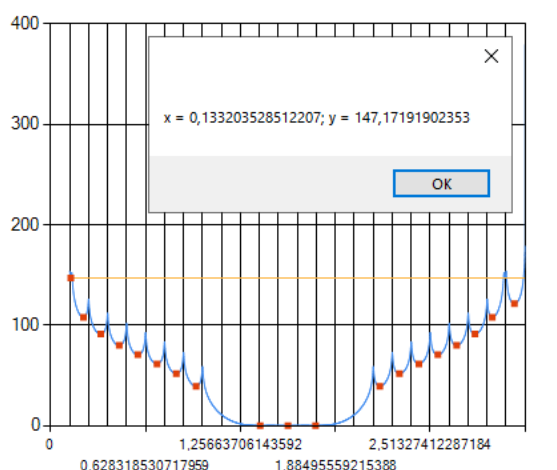


Рисунок 3.3 — Один з випадків неправильного знаходження гарантованого затухання

Далі у нас залишаються ті точки, з якими можна працювати. І їх групуємо в пари у клас `GetTwoPointWithLenghtBetweenList`. Він зберігає ліву точку, праву точку, і різницю висот між ними, яка знадобиться пізніше.

Як видно з графіка – він симетричний. І так і напрошується не розглядати, наприклад, праву частину, оскільки все можна знайти на лівій. Так і зроблено. Береться лише половина точок.

Зараз задача – знайти мінімальну точку в полосі затухання. Але серед цих точок що ми відібрали (половина графіку) справа також є декілька точок полоси пропускання. І ось вони і відсіюються за рахунок різниці висот між двома точками. Було зроблено спостереження, що різниця висот між останньою точкою зони затухання і першою точкою зони пропускання – найбільша. Знайшовши таку точку, легко відсіяти зону пропускання.

І ось, маючи точки, кожна з яких відповідає тенденції до спадання, просто береться остання з цих точок – і ось, вона лежить на лінії гарантованого затухання.

3.3 Знаходження найкращого рішення

В програмі реалізовано два підходи до знаходження найкращого рішення. Це звичайний – перебором, і генетичний. Звичайний перебір знаходиться з деякою точністю. Експерименти показали, що з точністю 0.01 результат знаходиться рівно за 2 хвилин. І значення гарантованого затухання – 70 децибел. А з 0.001 – за 8. І результат – 75 децибел. Як видно, різниця в 1 знак після коми вже може значно збільшити результат.

Перебір був потрібний, і буде для перевірки точності генетичного алгоритму. Крім того, можна прискорити виконання перебору якщо розподілити обчислення між потоками процесора.

Проте перебором не можна знайти найкраще рішення. Як мінімум через те, що крок в переборі фіксований, а найкращі результати можуть бути сконцентровані в деякі точці. Це можна побачити на рисунку 3.4. Зліва – область визначення точок при

переборі. Справа – область визначення точок при використанні генетичного алгоритму. X і Y – значення першого і другого коефіцієнтів. Вони знаходяться між 0 і 1.

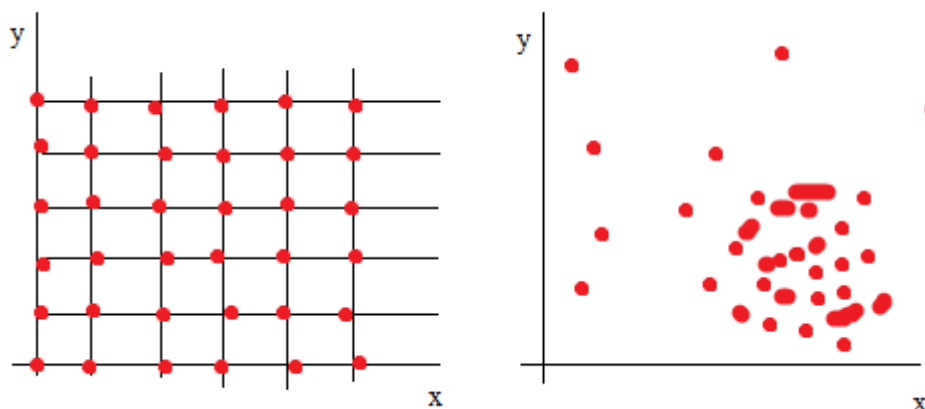


Рисунок 3.4 — Порівняння області визначення при переборі і при генетичному алгоритму

Якщо ж говорити за генетичний алгоритм, то він може швидше прийти до результату. Може хоч за 1 ітерацію. А може й за 30. Але одне точно – з його то точністю з якою він знаходить коефіцієнти, в нього більше шансів знайти кращий результат. Адже особливість функції така, що в параметрів висока чутливість. Тобто, однієї соті зовсім недостатньо для того аби перебрати всі можливі варіанти.

А що ж пропонує генетичний алгоритм для збільшення точності коефіцієнтів? Змінювати довжину гена. Під час експериментів з довжиною 32 біта, було знайдено значення 79.9 як видно на рисунку 3.5 :

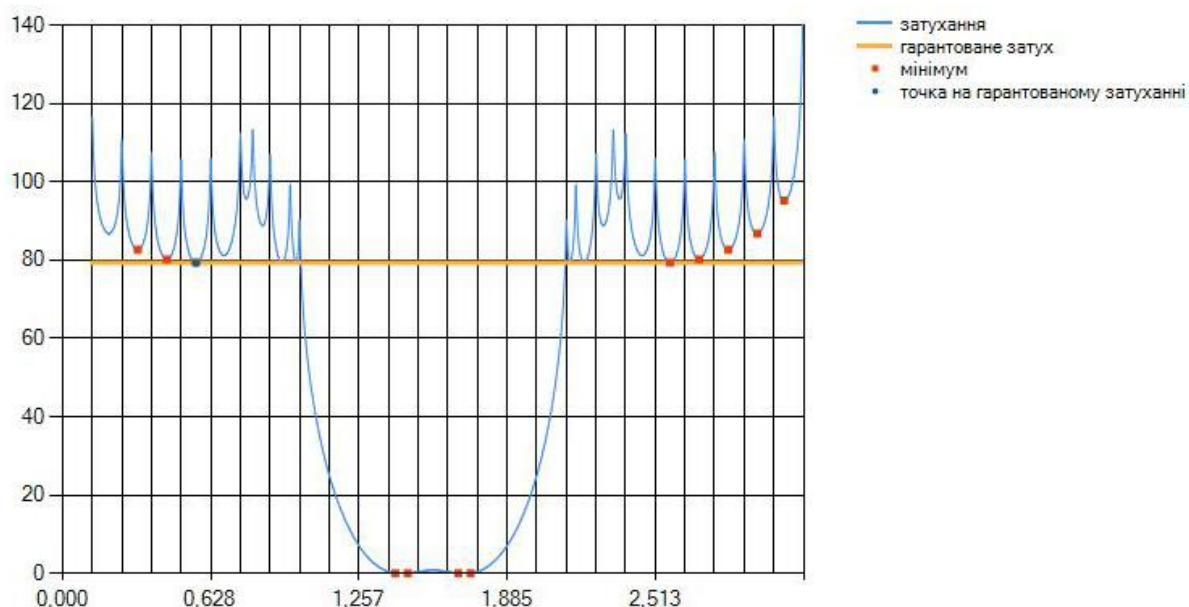


Рисунок 3.5 — Максимальне знайдене експериментами значення функції

Проте цей результат – не остаточний. Адже генетичний алгоритм містить високий відсоток удачі.

3.4 Алгоритм роботи генетичного алгоритму

На початку генерується стартова популяція особин. Далі з них підбираються індекси тих пар, які будуть брати участь у схрещуванні. Алгоритм такий : в циклі за допомогою випадкових чисел знаходять кандидатів для вибору батька. Зараз це одна третя від популяції. Тобто якщо в цій одній третій немає найкращих, то гірші особини мають шанс на виживання. Далі під час вибору батьків, перевіряється чи не одна й та сама особина є двома батьками, і чи не було вже такої пари раніше.

Далі на основі обчислених індексів формується діти. І батьки і діти беруть участь у визначенні нової популяції. Сортування за фітнес-функцією. Тобто, батьки мають шанс пройти далі, а діти можуть і не вижити.

Але перед тим як перейти до наступної ітерації потрібно ще зробити мутацію. Мутації може бути піддана будь-яка хромосома. Тобто, слабка особина може стати сильнішою.

Під кінець ітерації знаходиться найкраща особина. І якщо різниця між найкращими особинами поточної і минулої ітерації мінімальна, запускається «супермутація». Коли зі старої популяції залишається лише якась частина, а на місце інших підставляються заново згенеровані особини.

3.5 Реалізація оновлення графіка гарантованого затухання

За замовчуванням, якщо не робити додаткових дій, то всі обчислення в Windows Forms виконуються в одному потоку (одиниця виконання програми). Це накладає деякі обмеження для користувача. Такі як швидкість взаємодії з інтерфейсом. Оскільки інтерфейс і програмні обчислення знаходяться в одному потоці, то і оновлюватись в один і той самий час може щось лише одне. Причому не має якогось перемикання між завданнями, аби рівномірно все виконувалось по черзі. Тут або виконуються всі обчислення, або намалюється графік. Одночасно – ні. Причому це дуже незручно, що під час обчислень блокується інтерфейс, і не можна нічого з ним зробити.

Було зрозуміло, що потрібно розділити інтерфейс і обчислення по різних потоках. Щоб одне не блокувало інше. Для цього, перш за все, було використано можливості багатопоточного програмування C#, а саме такого класу як Thread.

А коли існують різні потоки, їх потрібно синхронізувати якось між собою. Передавати повідомлення. Для цього було використано делегати і події.

Розглянемо на прикладі модуля генетичного алгоритму як це працює.

З боку інтерфейсу, при натисненні на кнопку обрахунку генетичного алгоритму викликається метод GeneticAlgoRun. У ньому лиш на деякий час блокується інтерфейс. При підготовчій роботі до обчислень. А далі виконання генетичного алгоритму переходить в інший потік. В метод GeneticCalculations. В якому в свою чергу відкривається ще один потік для обрахунків. Далі в цьому ж методі в циклі перевіряються результати обрахунків. Якщо щось змінилось, то викликається метод в класі інтерфейсу для перемальовування інтерфейсу.

Крім принципу подій, між потоками інтерфейсу і модулем генетичного алгоритму йде передача інформації через статичні змінні.

3.6 Використання бази даних в програмі

Для оптимізації роботи користувача було вирішено зберігати результати роботи програми. Це не тільки дає можливість постійно не знаходити результати для однакових наборів вхідних даних, але й дає можливість використати це в генетичному алгоритмі. Яким саме чином? Мати чіткий кінець ітераціям, орієнтуючись що робота має закінчитись коли буде отримано результат кращий, за той що в сховищі даних .

Для зберігання даних використовується бінарний файл `database.bat`. Предметна область класів поповнилась такими класами як `BinaryDatabase` і `BinaryDatabaseEntity`. Очевидно, що в базу даних пишеться і читається перший клас. І він є місцем зберігання сутностей другого класу.

Об'єкти в файлі лежать за принципом так званої мапи. Коли є ключ і значення. Тобто для одного ключа є лише одне значення. В даному випадку ключем виступають параметри (порядок фільтра і т.д.) і коефіцієнти (проте не враховуючи тих двох які шукаються).

База даних за замовчуванням використовується після кожного обчислення. Чи то перебір, чи то генетичний алгоритм. Програма пробує зчитати дані з бази, і якщо для конкретних параметрів нічого немає, або поточне значення більше, тоді результат записується в базу даних, і при цьому видаляється старе значення.

Але користувач може й сам зчитати з бази значення, правда лише по поточним параметрам, що він ввів. Але й це може бути корисно.

Приклад вмісту сховища даних можна побачити на рисунку 3.6 :

Значення з бази даних :

	Коефіцієнт
▶	0,006897
	0,332901
	1
	1
	1
	1
	0,332901
	0,006897
*	

Гарантоване затухання :

55,068767

Взяти значення з бази даних

Рисунок 3.6 — Приклад вмісту бази даних

Висновок до розділу 3

Для реалізації знаходження коефіцієнтів в програмі було використано генетичний алгоритм. Один з плюсів даного вибору, це без сумніву те, що даний алгоритм вже добре досліджений. Існує багато реалізацій. Тому було потрібно лише адаптувати його під власні потреби.

Під час реалізації було знайдено деякі цікаві особливості. Наприклад, для прискорення обчислень застосовувалась «супермутація».

4 ОПИС ВИКОРИСТАНИХ ПРОГРАМНИХ ЗАСОБІВ

Для розробки такої системи було вирішено обрати мову C# через те, що з її бібліотекою Windows Forms можна доволі швидко отримати програму з інтерфейсом. В якості текстового редактора використано стандартне рішення – Visual Studio.

4.1 Мова програмування C#

C# - це мова програмування створена спеціально для роботи у середовищі Microsoft .NET Framework.

Мова C# була розроблена з урахуванням сильних і слабких особливостей інших мов, зокрема Java і C++. Специфікація мови C# була написана Андерсом Гейлсбергом, Скотом Вілтамутом та Пітером Гольде. Андерс Хейлсберг відомий у світі програмування як автор компілятора Turbo Pascal і лідер команди, яка створила Delphi.

Ключові особливості мови C#:

- Компонентна орієнтованість.
- Код зібраний воедино (декларації і реалізації об'єднані разом).
- Уніфікована система типів і їх безпечність.
- Автоматична і мануальна робота за пам'яттю.
- Використання єдиної бібліотеки класів – CLR.

До числа принципово важливих рішень, які реалізовані корпорацією Microsoft у мові програмування C #, можна віднести наступні:

- компонентно-орієнтований підхід до програмування (який характерний і для ідеології Microsoft. NET в цілому);
- властивості як засіб інкапсуляції даних (характерно також в цілому для ООП);
- обробка подій (наприклад, виключень);
- делегати;

— індексатори.

4.2 Технологія Windows Forms

Windows Forms - інтерфейс програмування додатків (API), що відповідає за графічний інтерфейс користувача і є частиною Microsoft .NET Framework.

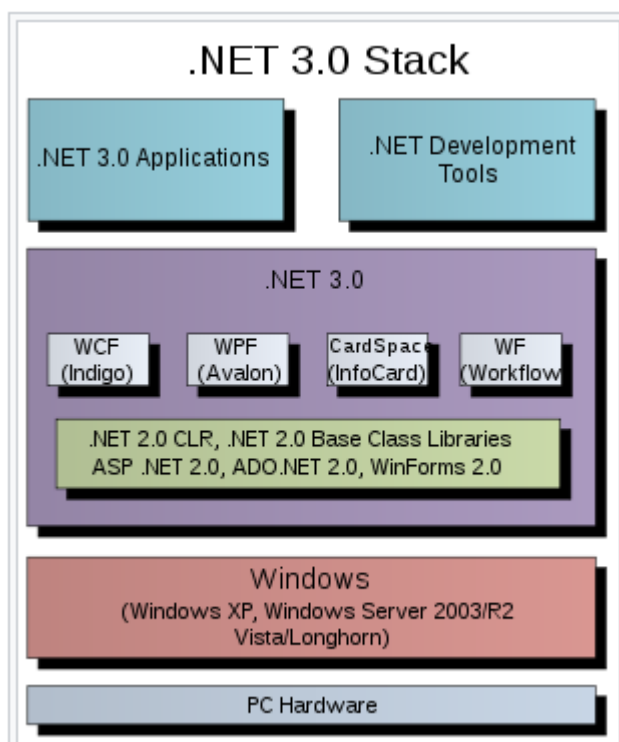


Рисунок 4.1 — Стек .Net Framework

Даний інтерфейс спрощує доступ до елементів інтерфейсу Microsoft Windows за рахунок створення обгортки для існуючого Win32 API в керованому коді. Причому керований код - класи, що реалізують API для Windows Forms, що не залежать від мови розробки. Тобто програміст однаково може використовувати Windows Forms як при написанні ПЗ на C #, C ++, так і на VB.Net, J # і ін.

З одного боку, Windows Forms розглядається як заміна більш старої і складної бібліотеки MFC, спочатку написаної на мові C ++. З іншого боку, WF не пропонує парадигму, порівнянну з MVC. Для виправлення цієї ситуації і реалізації даної функціональності в WF існують сторонні бібліотеки. Однією з найбільш

використовуваних подібних бібліотек є User Interface Process Application Block, випущена спеціальною групою Microsoft, що займається прикладами і рекомендаціями, для безкоштовного скачування. Ця бібліотека також містить вихідний код і навчальні приклади для прискорення навчання.

4.3 Середовище розробки Visual Studio

Інтегроване середовище розробки Visual Studio — це оригінальне середовище запуску, яке дозволяє редагувати, налагоджувати і створювати код, а потім публікувати додатки. Інтегроване середовище розробки (IDE) — це багатофункціональна програма, яку можна використовувати для різних аспектів розробки програмного забезпечення. Крім стандартного редактора і відладчика, які існують в більшості середовищ IDE, Visual Studio включає в себе компілятори, засоби виконання коду, графічні конструктори і багато інших функцій для спрощення роботи.

Висновок до розділу 4

Для розробки програми було вирішено обрати мову C#. Це об'єктно-орієнтована мова, що дозволяє виконувати різноманітні задачі. Також доволі важливим є те, що ця мова розвивається і сьогодні. Тобто є підтримка, зв'язок з користувачем. Тобто завжди можна знайти рішення можливої проблеми.

У якості фреймворку для розробки було вирішено вибрати Windows Forms. Це не найновіше рішення, адже існує ще WPF. Але Windows Forms — це надійний, перевірений з часом спосіб розробки десктоп-програм з інтерфейсом. В якості текстового редактора використано стандартне рішення — Visual Studio. Воно є достатньо зручним для розробника. І також як і мова — підтримується, розробляється і отримує нові рішення.

5 МЕТОДИКА РОБОТИ КОРИСТУВАЧА З ПРОГРАМНОЮ СИСТЕМОЮ

5.1 Системні вимоги та інсталяція

Для запуску програми потрібно мати операційну систему Windows 7/8/10. Процесор бажано з декількома ядрами (це підвищить швидкість обчислень). А оперативна пам'ять може бути різною в залежності від параметрів з якими ви бажаєте знаходити результат.

5.2 Сценарій роботи користувача з системою

При запуску програми користувач бачить таке вікно, як на рисунку 5.1 :

The screenshot shows a software interface with a menu bar at the top containing: 'Налаштування параметрів' (1), 'Н10' (2), 'Н18' (3), 'Н111' (4), 'АЧХ' (5), and 'Затухання' (6). Below the menu, there are several input fields and buttons:

- 'Порядок фільтра : 7' (7) with a text box containing '50'.
- 'Кількість розрядів квантування : 8' (8) with a text box containing '8'.
- 'k (потрібно для розрахунку X) : 1' with a text box containing '1'.
- 'Зміщення в В1К : 9' with a text box containing '9'.
- A button 'Побудова графіків' (8) is located to the right of the input fields.
- A button 'Очистити таблицю' (10) is located below the input fields.
- A table of coefficients is shown at the bottom right:

Коефіцієнт
0.3
0.6
1
1
1
1
0.6
0.3

The table has a red '9' at the bottom right corner.

Рисунок 5.1 — Стартове вікно

Верхня частина вікна містить кнопки переходу на закладки. (1) – налаштування параметрів, (2), (3), (4), (5) – переглянути графіки Н10, Н18, Н111 і АЧХ відповідно. А от по (6) знаходиться дві закладки.

Нижче зліва знаходяться параметри, які може ввести користувач. (7) – порядок фільтра, (8) – кількість розрядів квантування. І по кнопці «побудова графіків» - побудуються графіки. Побачити їх можна буде перейшовши на відповідну вкладку.

Коефіцієнти можна побачити і відредагувати на (9). Треба розуміти, що вводити треба симетрично їх. Наприклад, якщо зверху 0.3 і 0.6, то знизу – 0.6 і 0.3, як на рисунку 5.1.

По кнопці (10) таблиця коефіцієнтів очищається.

Перш ніж отримати гарантоване затухання, користувач має змогу побачити проміжні графіки, такі як Н10, Н18, Н111 і АЧХ.

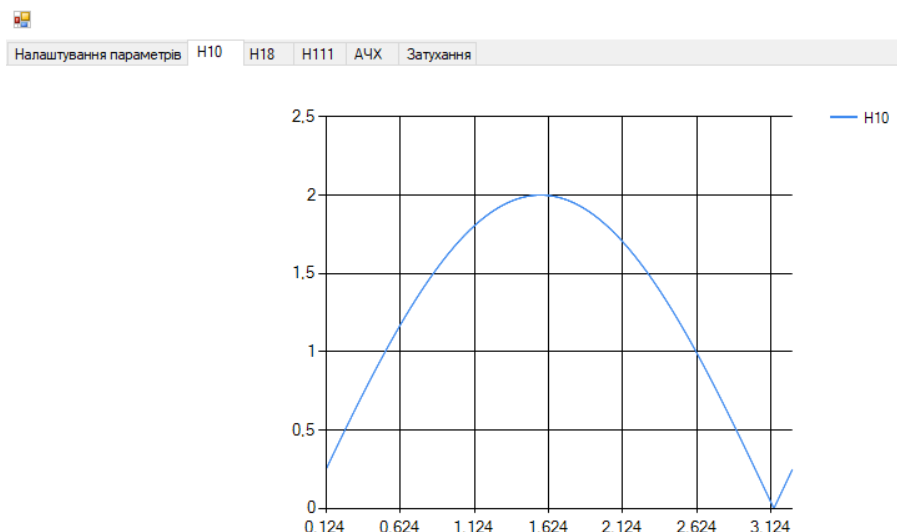


Рисунок 5.2 — Демонстрація проміжного графіка Н10

Н10, Н18, Н111 – це графіки, добуток яких формує функцію затухання. На рисунку 5.3 можна побачити графік Н18 :

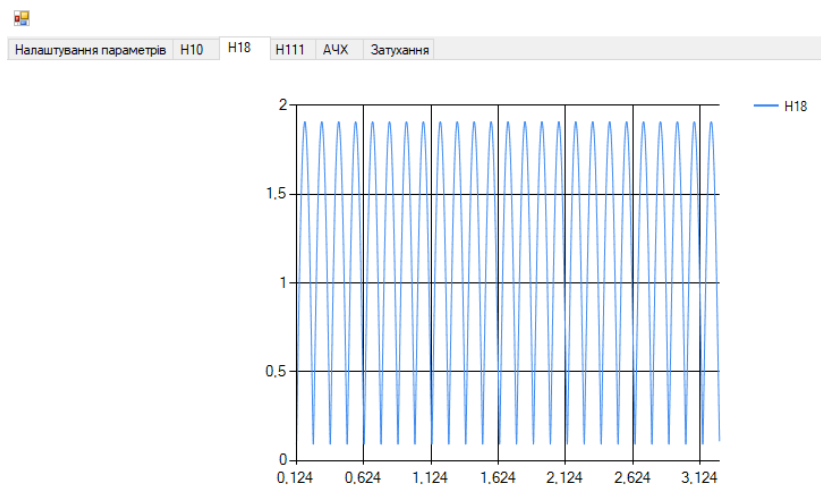


Рисунок 5.3 — Демонстрація проміжного графіка H18

А на рисунку 5.4 – H111.

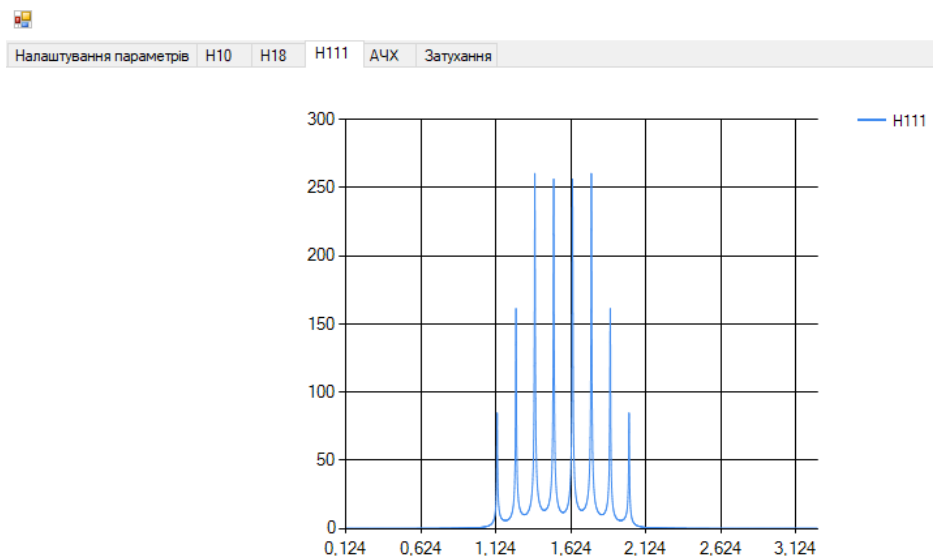


Рисунок 5.4 — Демонстрація проміжного графіка H111

Добуток H10, H18 і H111 формує деяку функцію АЧХ. А через АЧХ знаходиться функція затухання за допомогою логарифмування.

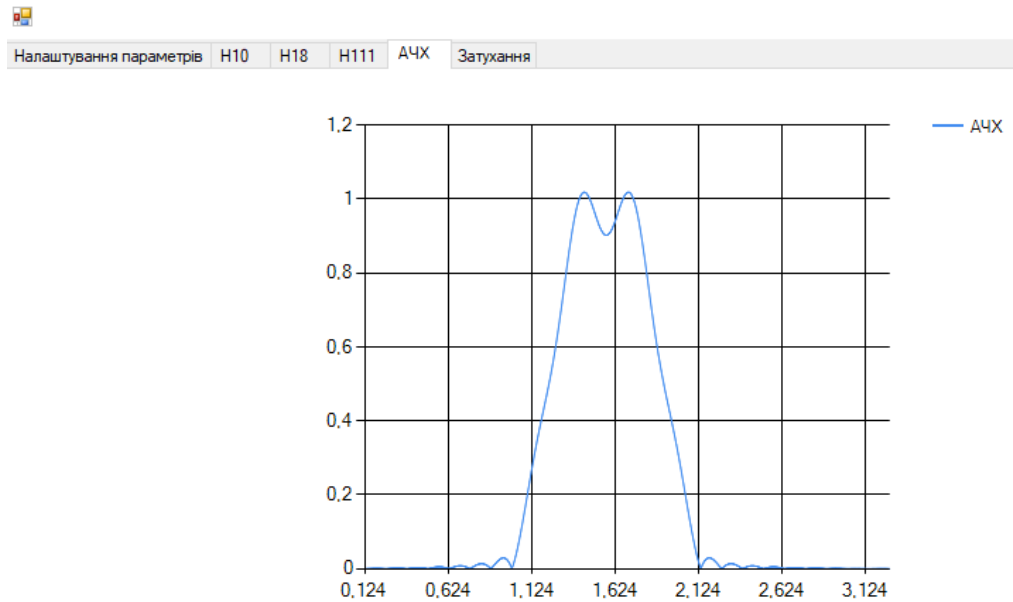


Рисунок 5.5 – Демонстрація АЧХ

В генетичному алгоритмі можна вказати багато параметрів його роботи :

Налаштування параметрів Н10 Н18 Н111 АЧХ Затухання

Налаштування Графік

Коефіцієнт
*

Налаштування генетичного алгоритму :

максимальна к-сть ітерацій : 30

розмір популяції : 60

к-сть генів в хромосомі : 16

% осіб що розглядаєм при виборі батька : 30

% мутованих осіб : 30

% мутації гена при мутації : 50

різниця Y для того аби зробити "жостку" мутацію : 0,000001

поточне гарантоване затухання

Знайти рішення перебором

знайти рішення генетичним алгоритмом

Рисунок 5.6 — Меню затухання

На рисунку 5.6 видно меню затування. Область (1) – закладка налаштування. Тут можна вибрати один з трьох варіантів розвитку подій : або обчислити гарантоване затування за тими параметрами, що були задані в попередньому вікні, або вибрати повний перебір(6) (з деякою точністю, за замовчуванням 0.01), або генетичний алгоритм(7), який можна налаштувати за допомогою області (4).

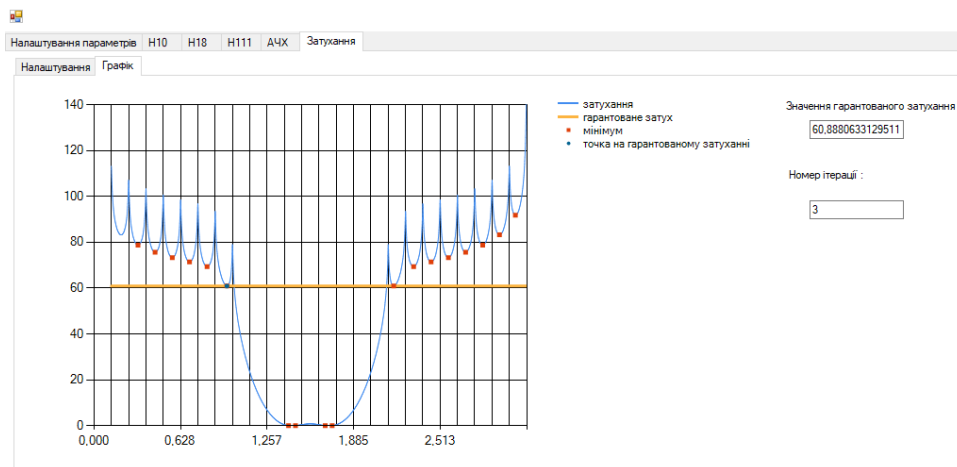


Рисунок 5.7 — Вікно з графіком затування

При переході на (3) користувача чекає вікно, де при генетичному алгоритмі в реальному часі змінюється графік. І можна побачити номер ітерації, і поточне максимальне значення.

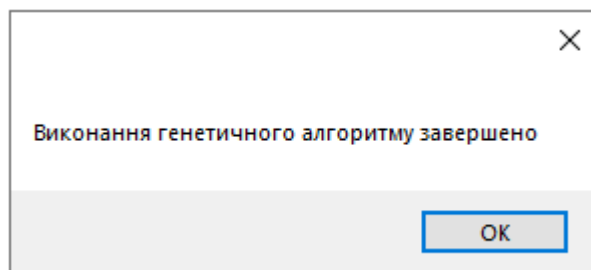


Рисунок 5.8 — Повідомлення після виконання генетичного алгоритму

Після виконання з'являється інформаційне вікно. Наприклад, на 5.8 видно вікно після роботи генетичного алгоритму.

В програмі існує власна база даних. Інтерфейс перегляду вмісту бази даних показано на 5.9.

Налаштування параметрів Н10 Н18 Н111 ДЧХ Затухання

Налаштування Графік

Коефіцієнт
*

Налаштування генетичного алгоритму :

максимальна к-сть ітерацій : 30

розмір популяції : 60

к-сть генів в хромосомі : 16

% осіб що розглядаєм при виборі батька : 30

% мutowаних осіб : 30

% мутації гена при мутації : 50

різниця Y для того аби зробити "жорстку" мутацію : 0.000001

поточне гарантоване затухання

Знайти рішення перебором

Знайти рішення генетичним алгоритмом

Значення з бази даних :

Коефіцієнт
*

Гарантоване затухання :

Взяти значення з бази даних

Рисунок 5.9 — Порожнє вікно перегляду значень бази даних (справа)

Якщо ж користувач натисне на кнопку (1), тоді в базі даних буде відбуватись пошук значення для поточних параметрів і коефіцієнтів. Якщо такі будуть знайдені – то порожня таблиця і поле заповняться значеннями. На рисунку 5.10 можна це побачити :

Значення з бази даних :

Коефіцієнт
0,006897
0,332901
1
1
1
1
0,332901
0,006897
*

Гарантоване затухання :

55,068767

Взяти значення з бази даних

Рисунок 5.10 — Приклад отриманих значень з бази даних

Висновок до розділу 5

Розроблена програма враховує потреби користувача. Тому дозволяє задати всі потрібні параметри для обчислень. Також вона демонструє всі етапи обчислень, що корисно для виявлення можливих проблем.

Стосовно реалізації генетичного алгоритму – у програмі є зручний динамічний режим зміни графіку. В реальному часі видно як програма через ітерації приходить до потрібного результату.

Також присутнє збереження результатів. Це означає що не потрібно постійно перераховувати значення, якщо для поточних параметрів вони вже були колись знайдені. Але з іншого боку, якщо знайдені генетичним алгоритмом – то не факт що це найкраще значення. Адже кінцевого ідеального рішення там не існує, на відміну від перебору, де задається конкретна точність і відома кількість рішень, і залишається їх просто перебрати.

6 РОЗРОБКА СТАРТАП ПРОЕКТУ

6.1 Опис ідеї проекту

У даному розділі описано економічне обґрунтування реалізації стартап-проекту на тему «Оптимізація коефіцієнтів цифрового фільтру за допомогою генетичного алгоритму».

Користувачу пропонується отримати програму, які можна використати для фільтрації сигналів методом частотної вибірки.

Цільова аудиторія програми – спеціалісти в області проектування цифрових фільтрів.

Переваги використання даної програми – її швидкість і краще рішення, ніж в аналогів що використовують частотну вибірку також. Оскільки в даній програмі використовується генетичний алгоритм. Аналоги можуть зупинитись на не найкращому рішенні, а наш продукт – ні.

Звичайно ж, для створення стартапу потрібна ідея, оцінка ризиків, порівняння з конкурентами, оцінка сильних і слабких сторін. Це допоможе оцінити можливості продукту на ринку і знайти потенційних клієнтів. Саме це і буде описано в розділі.

Проект починається з ідеї. Яку можна побачити в таблиці 6.1 :

Таблиця 6.1. Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
1	2	3
Система для оптимізації коефіцієнтів цифрового фільтру за допомогою генетичного алгоритму	Розробка цифрових фільтрів	Швидкодія, знаходження ефективного результату

Продовження таблиці 6.1

1	2	3
	Освітній процес	Демонстрація роботи генетичного алгоритму, Демонстрація знаходження коефіцієнтів цифрового фільтра

Як видно з таблиці 6.1 – основний напрям застосування – розробка цифрових фільтрів. Саме таких фільтрів – з частотною вибіркою і генетичним алгоритмом – на ринку не існує. Є конкуренти, які використовують інші підходи – це є t-filter.engineerjs і falstad.

В розробці цифрових фільтрів користувачам важливо мати швидке і якісне рішення. Цю потребу може задовольнити даний продукт.

Але також даний стартап-проект може бути корисним у освітньому процесі. Для того аби показати як формуються графіки в цифровій фільтрації, і як працює генетичний алгоритм. За рахунок динамічної зміни графіка за цим дуже зручно спостерігати.

Наступним кроком є визначення характеристик продукту, що планується розробити (таблиця 6.2).

З таблиці 6.2 зрозуміло, що відбувається порівняння з деякими конкурентами.

Таблиця 6.2. Визначення сильних, слабких та нейтральних характеристик ідеї проекту

№ п/п	Техніко- економіч ні характери стики ідеї	(потенційні) товари/концепції конкурентів			Слабка сторона	Нейтральна сторона	Сильна сторона
		Розроблюваний проект	Конкурент 1	Конкурент 2			
1	Точність	висока	висока	висока			+
2	Собі- вартість	низька	середня	висока		+	
3	Кросплат форменні сть	Ні	Так	Ні	+		
4	Швидкод ія	Так	Так	Ні			+
5	Надій- ність	Необме- жений термін дії	Необме- жений термін дії	Необмежений термін дії		+	
6	Збережен ня результат ів	Так	Так	Так			+

На основі цих міркувань можна стверджувати конкурентоспроможність продукту.

6.2 Технологічний аудит ідеї проекту

В цьому пункті слід провести аудит технології проекту, що має допомогти реалізувати ідею (технології створення товару).

Крок для цього – це знайти технології, які можуть використовуватись в процесі розробки (таблиця 6.3) :

Таблиця 6.3. Технологічна здійсненність ідеї проекту

Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технологій
Розробка програмного забезпечення для оптимізації коефіцієнтів передавальної функції цифрового фільтру за допомогою генетичного алгоритму	Технологія 1. Самостійна розробка всіх компонентів програми без використання сторонніх бібліотек на мові C#.	Потребує розробки	Немає у відкритому доступі
	Технологія 2. Розробка компонентів програми з використанням функціоналу існуючих відкритих бібліотек (Windows Forms).	Наявні	У відкритому доступі

З наведеної таблиці видно, що основний функціонал проекту потрібно реалізувати, а допоміжний функціонал можна отримати за допомогою фреймворку Windows Forms.

6.3 Аналіз ринкових можливостей запуску стартап-проекту

Визначення ринкових можливостей, які можна використати під час ринкового впровадження проекту, та ринкових загроз, які можуть перешкодити реалізації проекту, дозволяє спланувати напрями розвитку проекту із урахуванням стану ринкового середовища, потреб потенційних клієнтів та пропозицій проектів-конкурентів.

Спочатку варто провести аналіз попиту (таблиця 6.4).

Таблиця 6.4. Попередня характеристика потенційного ринку стартап-проекту

№ п/п	Показники стану ринку	Характеристика
1	Кількість головних гравців, од	Біля 3
2	Загальний обсяг продаж, грн/ум.од	10000 за рік
3	Динаміка ринку (якісна оцінка)	Очікується зростання
4	Наявність обмежень для входу (вказати характер обмежень)	Немає
№ п/п	Показники стану ринку	Характеристика
5	Специфічні вимоги до стандартизації та сертифікації	Немає

Таблиця 6.5. Характеристика потенційних клієнтів стартап-проекту

Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
Оптимізація коефіцієнтів перехідної функції цифрового фільтру за допомогою генетичного алгоритму	Науковці	Використання в процесі дослідження розробки цифрових фільтрів	Швидкість роботи, точність результатів, отримання результатів потрібних для дослідження сигналів
	Студенти	Використання в процесі вивчення курсу цифрової фільтрації	Зрозумілий і простий інтерфейс, точність результатів, низька ціна
	Викладачі	Викладання курсу цифрових сигналів	Зрозумілий і простий інтерфейс, точність результатів
	Розробники цифрових фільтрів	Використання програми для підбору коефіцієнтів	Швидкість роботи, точність результатів, отримання результатів потрібних для дослідження сигналів

Як видно з таблиці 6.5 основними користувачами продукту є розробники цифрових фільтрів. Для них важливою є швидкість роботи, точність результатів. Крім того продукт є корисним для студентів і викладачів для вивчення цифрової фільтрації.

Наступним кроком є аналіз ринкового середовища. Необхідно в табличному вигляді визначити фактори загроз (таблиця 6.6) і можливостей (таблиця 6.7), що перешкоджають і допомагають впровадженню програми.

Таблиця 6.6. Фактори загроз

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
1	Збільшення ціни на розробку	Збільшення витрат на розробку, розширення технологічної бази.	Зміна цінової політики, додавання ексклюзивного функціоналу, перехід на більш перспективні та ефективні технології
2	Поява нових гравців на ринку	Поява на ринку конкурентів з аналогічними продуктами	Реклама, удосконалення продукту, додавання ексклюзивного функціоналу.

Таблиця 6.7. Фактори можливостей

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
1	Попит на програмний продукт	Потреба у якісному аналізі гідроакустичних сигналів сприяє збільшенню попиту на відповідне ПЗ.	Розробка та підтримка унікального нового надійного функціоналу
2	Зменшення ціни на розробку	Ефективніше використання наявних ресурсів.	Збільшення штату компанії, збільшення швидкості розробки.

Наступним кроком є визначення типу конкуренції (таблиця 6.8).

Таблиця 6.8. Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
1. Тип конкуренції - олігополія	Домінує мала кількість компаній	Розробка унікального функціоналу, потужна рекламна кампанія.
2. За рівнем конкурентної боротьби: конкурентне середовище міжнародне	Боротьба ведеться на міжнародному ринку	Локалізація продукту, розвиток рекламної кампанії.
3. За галузевою ознакою - внутрішньогалузева	Продукт використовується у сфері розробки цифрових фільтрів	Розробка унікального функціоналу, щоб повністю задовольнити користувачів
4. Конкуренція за видами товарів: - товарно-родова	Конкуренція між різними видами систем розробки цифрових фільтрів.	Розробка унікального функціоналу.
5. За характером конкурентних переваг - не цінова	Ціна не є основним фактором конкуренції. Конкурентні переваги продукту засновані на його функціональних можливостях.	Впровадження нових функціональних можливостей

Продовження таблиці 6.8

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
6. За інтенсивністю - не марочна	Продукт прив'язаний до відповідних розробок компанії (MatLab, Mathematica)	Інтеграція нових можливостей та різних систем.

Такий аналіз конкуренції є неповним. Отже треба провести аналіз конкуренції в галузі за М. Портером (таблиця 6.9).

Таблиця 6.9. Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
	Відсутні	Присутні	Відсутні	Споживачі диктують умови.	Присутні, зокрема Конкуренти 1,2.
Висновки	Прямі конкуренти на ринку відсутні.	Присутні. Вихід на ринок можливий.	Постачальники відсутні, потрібні спеціалісти сфери розробки ПЗ.	Клієнти диктують умови. Якісний, дешевий, швидкодіючий продукт.	Обмежень для роботи на ринку товари-замінники не представляють

Згідно таблиці, зрозуміло що робота на ринку можливо. Ринку потрібна система даного типу.

Використовуючи ці висновки, можна виокремити основні фактори конкурентоспроможності продукту (таблиця 6.10).

Таблиця 6.10. Обґрунтування факторів конкурентоспроможності

№ п/п	Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)
1	Швидкість обчислень	Аналіз проводиться за допомогою новітніх методів, що значно пришвидшує роботу.
2	Різносторонній аналіз	Система одночасно проводить різносторонній аналіз сигналу, що сприяє визначенню прихованих кореляцій.
3	Точність результатів	Висока точність результатів за рахунок правильного підходу до аналізу сигналі

Після таких висновків вже можна проаналізувати сильні та слабкі сторони проекту у порівнянні з конкурентами (таблиця 6.11).

Таблиця 6.11. Порівняльний аналіз сильних та слабких сторін проекту

№ п/п	Фактор конкурентоспроможності	Бал	Рейтинг товарів-конкурентів						
			-3	-2	-1	0	+1	+2	+3
1	Швидкість аналізу	20			K2	K1			П
2	Різносторонній аналіз	19				K2	K1	П	
3	Точність результатів	18				K1	K2	П	
4	Збереження результатів	17				П	K1	K2	
5	Доступність	15				K2	K1		П

SWOT-аналіз (матриця аналізу сильних (Strength) та слабких (Weak) сторін, загроз (Troubles) та можливостей (Opportunities) на основі виділених ринкових загроз та можливостей, та сильних і слабких сторін приведено в таблиці 6.12.

Таблиця 6.12. SWOT-аналіз стартап-проекту

Сильні сторони: швидкість аналізу, точність аналізу, можливість збереження результатів.	Слабкі сторони: доступність
Можливості: збільшення попиту на товар, зменшення коштів на розробку.	Загрози: збільшення ціни на розробку, нові гравці на ринку

На основі цього аналізу розроблено альтернативи ринкової поведінки для просування стартапу на ринок та приблизний оптимальний час їх ринкової реалізації з огляду на потенційні проекти конкурентів.

Використовуючи SWOT-аналіз, можна скласти таблицю 6.13 альтернатив ринкового впровадження стартап-проекту.

Таблиця 6.13. Альтернативи ринкового впровадження стартап-проекту

№ п/п	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1	Розробка базової версії продукту, з забезпеченням запланованого функціоналу	висока	До 2 місяців

Продовження таблиці 6.13

№ п/п	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
2	Розробка версії продукту з використанням власної системи аналізу даних.	Низька	До 6 місяців
3	Створення онлайн версії продукту	Середня	До 3 місяців
4	Впровадження найновіших ефективних технологій	низька	До 5 місяців

Обрана альтернатива перша, оскільки її реалізація є найбільш вірогідна та з малими строками.

6.4 Розробка ринкової стратегії проекту

В розробленні ринкової стратегії перше, що треба зробити, це визначити стратегію охоплення ринку (таблиця 6.14).

Таблиця 6.14. Вибір цільових груп потенційних споживачів

Опис профілю цільової групи потенційних клієнтів.	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
Інститути та університети, що займаються розробкою цифрових фільтрів	Споживачі готові сприйняти продукт.	Попит високий в цільовому сегменті	Конкуренція з товарами-замінниками	Продукт простий у впровадження його на ринок

Обрана стратегія концентрованого маркетингу, оскільки присутній один цільовий сегмент.

На основі цих міркувань можна навести базову стратегію розвитку (таблиця 6.15).

Таблиця 6.15. Визначення базової стратегії розвитку

Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
Розробка базової версії функціоналу, з подальшим покращенням.	Стратегія розширення первинного попиту	Конкурентами є лише товари-замінники. Розроблювана система має переваги в швидкості, точності та простоті використання.	Стратегія диференціації

Наступний крок — визначення базової стратегії конкурентної поведінки (таблиця 6.16).

Таблиця 6.16. Визначення базової стратегії конкурентної поведінки

Чи є проект «першопроходцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки*
Ні	Шукати нових та забирати споживачів у конкурентів	Ні	Стратегія лідера

З даної таблиці видно, що стартап-проект не є першопроходцем на ринку. Ринок є в деякій мірі досліджений, отже ризики не максимальні. Але прийдеться придумати стратегію як забирати споживачів у конкурентів.

Серед стратегій буде вибрано стратегію лідера.

Зважаючи на вимоги користувачів, стратегії розвитку та конкурентної поведінки, далі треба розробити стратегію позиціонування. Під цією стратегією розуміється формування ринкової позиції (комплексу асоціацій), що допоможе споживачам з ідентифікацією проекту (таблиця 6.17).

Таблиця 6.17. Визначення стратегії позиціонування

№ п/п	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап-проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту (три ключових)
	Швидкість, точність, унікальний функціонал	Стратегія диференціації	Швидкість, точність, простота використання	Швидкість, точність, сучасність

6.5 Розроблення маркетингової програми

Розробляючи маркетингову програму, з самого початку потрібно сформувати маркетингову концепцію товару (таблиця 6.18).

Таблиця 6.18. Визначення ключових переваг концепції потенційного товару

№ п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
1	Швидкодія	Швидкодія за рахунок використання генетичного алгоритму	Процес аналізу швидший ніж у конкурентів.
2	Точність	Точність за рахунок використання генетичного алгоритму	Диференціація експериментів за певними показниками та визначення найкращих по певному показнику.

Продовження таблиці 6.18

№ п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
3	Простота використання	Інтуїтивно зрозумілий неперевантажений інтерфейс	Конкурентні рішення мають доволі перевантажений та складний інтерфейс.

Серед перерахованих вище потреб, можна сказати що програма буде задовольняти потребу в швидкодії. Це буде зроблено в першу чергу за допомогою генетичного алгоритму.

Крім того, генетичний алгоритм дає високу точність, на відміну від тих алгоритмів, що використовуються в аналогах.

А стосовно простоти використання – інтерфейс є доволі зрозумілим, принаймні для розробників цифрових фільтрів. Крім того реалізовано динамічний показ того що відбувається з графіками.

Наступний крок — розробка трирівневої маркетингової моделі товару. Тут варто розглянути такі рівні товару, як задум і реальне виконання.

Таблиця 6.19. Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові
I. Товар за задумом	Програмне забезпечення для оптимізації коефіцієнтів цифрового фільтру за допомогою генетичного алгоритму. Можна виділити наступні вигоди від використання продукту: <ul style="list-style-type: none"> — Швидкість аналізу даних; — Точність; — Простота використання;

Продовження таблиці 6.19

Рівні товару	Сутність та складові		
II. Товар у реальному виконанні	Властивості/характеристики	М/Нм	Вр/Тх /Тл/Е/Ор
	1. Висока швидкодія	М	Тх/Тл
	2. Висока точність	М	Тх/Тл
	3. Різносторонній аналіз	М	Тх/Тл
	4. Простота використання	М	Вр/Тл
	Якість: відповідає нормам розробки програмного забезпечення.		
	Пакування: Продукт представлений у десктопного додатку. На сайті міститься: <ul style="list-style-type: none"> — загальна назва продукту, власна назва; — опис продукту; — функції, які представлені; інструкція для користування; <ul style="list-style-type: none"> — контакти для зв'язку з розробником; — підтримка для клієнтів. 		
	Марка: ПП «FSF Designer»		

Проект буде захищено як інтелектуальна власність. Товар буде захищено від копіювання та неправомірного розповсюдження за допомогою обфускації коду.

Далі потрібно визначити цінові межі (таблиця 6.20).

Таблиця 6.20. Визначення меж встановлення ціни

Рівень цін на товари-замінники	Рівень цін на товари-аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар
15 тис. грн	Відсутні	600 тис. грн	5 тис. грн – 7 тис. грн

Після цього необхідно визначити систему збуту, оптимальну для даного проекту (таблиця 6.21).

Таблиця 6.21. Формування системи збуту

Специфіка поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
Реєстрація на сайті, оплата продукту, відсилання на електрону пошту ключа активації.	Доставка на електрону пошту ключа активації.	Виробник - споживач	Web-сайт

Останньою складовою маркетингової програми є розроблення концепції маркетингових комунікацій, що спирається на попередньо обрану основу для позиціонування, визначену специфіку поведінки клієнтів. Результат можна побачити в таблиці 6.22.

Таблиця 6.22. Концепція маркетингових комунікацій

Специфіка поведінки цільових клієнтів	Канали комунікацій цільових клієнтів	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
Ретельно обирають систему	Веб-сайт, телефон, месенджер	Підтримка, індивідуальний підхід, постійне оновлення	Донести переваги продукту до клієнта	Оптимізація коефіцієнтів передавальної функції цифрового фільтру

Висновок до розділу 6

В даному розділі розглянуто основні моменти виходу на ринок стартап-проекту, що є десктоп-додатком для оптимізації коефіцієнтів передавальної функції цифрового фільтру за допомогою генетичного алгоритму.

Було запропоновано та детально розглянуто ідею для проекту, а також проведено технологічний аудит та аналіз ринкових можливостей запуску стартап-проекту. Після цього наступним кроком було розроблено ринкову стратегію проекту.

Крім того, було проаналізовано слабкі та сильні сторони проекту. Було проаналізовано можливі ризики та запропоновано необхідні дії в разі їх прояву. Враховуючи це, було розроблено SWOT-аналіз.

В ході розробки програми було детально порівняно розроблюваний проект з аналогічними програмами конкурентів, такими як t-filter та falstad. Порівняльний аналіз показав, що і в конкурентів, і в розроблюваній програмі є свої плюси та мінуси, проте в загальному остання має більш кращу і вигідну для користувачів сукупність показників.

Було виявлено, що існує можливість комерціалізації такого проекту, адже показники ринку цьому сприяють (наприклад, є попит на подібні товари, очікується зростання ринку у цій сфері). Враховуючи потенційні групи клієнтів, невисокі бар'єри входження, доведену вище конкурентоспроможність продукту, можна зробити висновок про існування перспектив впровадження такого проекту.

ВИСНОВКИ

В ході виконання роботи проаналізовано літературу стосовно проектування цифрових фільтрів з оптимальними параметрами. В результаті виявлено, що існує мало літератури по даній темі. А також розглянуто літературу стосовно методів глобальної оптимізації. Також модифіковано генетичний алгоритм для оптимізації коефіцієнтів передавальної функції цифрового фільтра.

Створено систему, що дозволяє знаходити оптимальні коефіцієнти цифрового фільтра під вказані користувачем параметри, використовуючи генетичний алгоритм. Система дозволяє побачити результати на графіках, і зберегти результати в базі даних.

Також обґрунтовано актуальність цієї розробки. Програма орієнтована на таких користувачів як розробники цифрових фільтрів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Рабинер Л. Теория и применение цифровой обработки сигналов / Л. Рабинер, Гоулд. – Москва, 1978.
2. Голд Б. Цифровая обработка сигналов / Б. Голд, Ч. Рэйдер. // «Сов. Радио». – 1973. – С. 368.
3. Айфичер Э. Цифровая обработка сигналов, Практический подход / Э. Айфичер, Б. Джервис. – Москва, 2004. – (Второе издание).
4. Whitley D. L. Evolution, The Baldwin Eectand Function Optimization .In : Pro c.Parallel Problem Solving from Nature / D. L. Whitley, V. S. Gordonand. – Berlin, Springer Verlag, 1994.
5. Goldberg D. E. Loci and the Traveling Salesman Problem In : Pro ceding sof International Conference on Genetic Algorithms and Their Applications / D. E. Goldberg, R. Lingle. – Lawrence, Hillsdale, 1985.
6. Віннічук М.В. Оптимізація коефіцієнтів цифрового фільтру методом частотної вибірки за допомогою генетичного алгоритму / М.В. Віннічук, І.А. Варава // Інформаційні технології – 2019: зб. тез VI Всеукраїнської науково-практичної конференції молодих науковців. – Київ. – 16 травня 2019.
7. Рибка С.В. Методика автоматизованого вирішення задачі апроксимації цифрових фільтрів на основі частотної вибірки / С.В. Рибка, А.П. Корольов, А.М. Мацаєнко, І.А. Варава // Збірник наукових праць ВІТІ. — 2019. — № 2. — С. 69—77. - Режим доступу: http://www.viti.edu.ua/files/zbk/2019/9_2_2019.pdf.
8. Bertsekas D. P. Parallel and Distributed Computation: Numerical Methods / D. P. Bertsekas, J. N. Tsitsiklis., 1989.
9. Dennis J. E. Direct Search Methods on Parallel Machines / J. E. Dennis, V. Torczon., 1991.
10. Hamma B. Parallel Continuous Simulated Annealing for Global Optimization / B. Hamma, S. Viitanen, A. Törn., 2000.

11. Nemhauser G. L. Branch-and-bound and Parallel Computation: a Historical Note / G. L. Nemhauser, E. A. Pruul., 1988.
12. Amdahl G. The Validity of Single Processor Approach to Achieving Large Scale Computing Capabilities / Amdahl., 1967.
13. Евтушенко Ю. Г. Численный метод поиска глобального экстремума функций (перебор на неравномерной сетке) / Ю. Г. Евтушенко., 1971.
14. Ali M. Stochastic Global Optimization: Problem, Classes and Solution Techniques / M. Ali, A. Törn., 1999.
15. Törn A. Global Optimization / A. Törn, A. Zilinskas. – Berlin, Springer Verlag, 1989.
16. Törn A. Topographical Global Optimization / A. Törn, S. Viitanen., 1992.
17. Evtushenko Y. G. Recent Advances in Global Optimization / Y. G. Evtushenko, M. A. Potapov., 1992.
18. Hansen E. R. Global Optimization Using Interval Analysis / Hansen. – New York, Marcel Dekker, 1992.
19. Moore R. Interval Analysis / Moore. – New Jersey, Prentice-Hall, 1966.
20. Nemhauser G. L. Branch-and-bound and Parallel Computation: a Historical Note / G. L. Nemhauser, E. A. Pruul., 1988.
21. Ratschek H. Interval Methods / H. Ratschek, J. G. Rokne., 1995.
22. Moore R. Interval Analysis / Moore. – New Jersey, Prentice-Hall, 1988.
23. Nemhauser G. L. Branch-and-bound and Parallel Computation: a Historical Note / G. L. Nemhauser, E. A. Pruul., 1988.
24. Gomez S. The Tunneling Method Applied to Global Optimization / S. Gomez, A. V. Levy., 1985.
25. Price W. L. Controlled Random Search Procedure for Global Optimization / Price. // The Computer Journal. – 1977. – №20. – С. 367–370.
26. Ali M. Numerical Comparison of Some Modified Controlled Random Search Algorithms / M. Ali, A. Törn., 1997.
27. Diener I. On the Global Convergence of Path-following Methods to Determine All Solutions to a System of Nonlinear Equations / Diener., 1987.

28. Diener I. Trajectory Methods in Global Optimization / Diener. – Dordrecht, Kluwer, 1995.
29. Sturua E. G. A Trajectory Algorithm Based on the Gradient Method. The Search on Quasioptimal trajectories / E. G. Sturua, S. K. Zavriev. // J. of Global Optimization. – 1991. – №1.
30. Zavriev S. K. On the Global Optimization Properties of Finite-difference Local Descent Algorithms / Zavriev. // J. of Global Optimization. – 1993. – №3. – C. 63–78.

ДОДАТОК А

Оптимізація коефіцієнтів цифрового фільтру методом частотної вибірки за допомогою генетичного алгоритму

Тези на конференцію «VI Всеукраїнська науково-практична конференція молодих науковців (м. Київ, 16 травня 2019р.)»

УКР.НТУУ"КПІ ім. Ігоря Сікорського" _ТЕФ_АПЕПС_TV4195_19М

Аркушів 2

Київ 2019

Магістрант Віннічук М.В., доцент Варава І.А.

Оптимізація коефіцієнтів цифрового фільтра методом частотної вибірки за допомогою генетичного алгоритму

Найчастіше цифрова фільтрація застосовується для обробки сигналу в системах, що оперують дискретними даними. Виділення "корисного" сигналу необхідно, коли сигнал, що надходить у систему із зовнішнього середовища, змішаний із шумами, викликаними різноманітними фізичними процесами, що мають, як правило, випадковий характер.

Цифрові фільтри задаються такими характеристиками як порядок, частота зрізу, гарантоване затухання, нерівномірність[1].

Що таке виділення «корисного» сигналу? Якщо відрізнити сигнал за частотою, то можна сказати, що частоти від точки X_1 до X_2 пропускаємо, реагуємо на них, обробляємо, а всі інші частоти – нижчі за X_1 , і вищі за X_2 – ігноруємо. Точки X_1 та X_2 ще називають частотами зрізу.

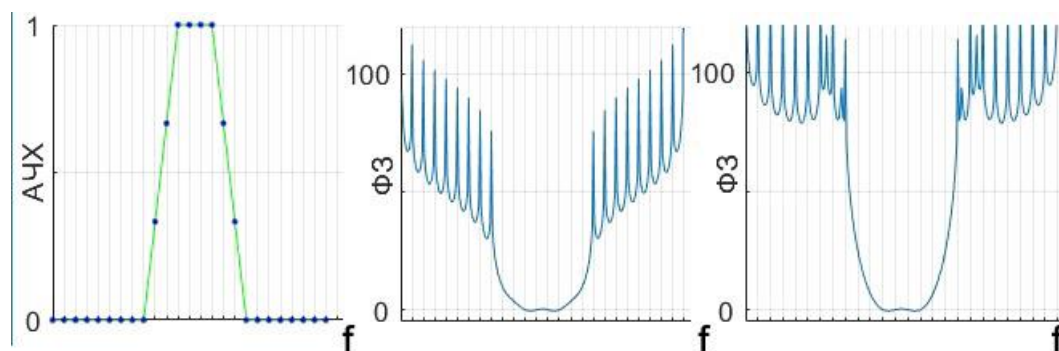


Рис.1 а – АЧХ фільтра, б, в – графіки функції затухання сигналу

На рисунку 1-а показано, що деяка смуга частот (f) пропускається. А на рисунку 1-б показано за рахунок чого. Показано графік затухання. І в ідеальному фільтрі локальні мінімуми в правій та лівій полосі затримування приймають якомога найбільше середнє значення з мінімальною дисперсією.

Проте це ідеальний фільтр. А в реальному житті графік виглядає з більшими нерівностями. Це можна побачити на 1-в. І проектування фільтра полягає в тому, або згладити дані нерівності.

При проектуванні фільтра стоїть задача забезпечити гарантоване затухання сигналу на частотах зрізу, і максимально прибрати нерівномірність в полосі пропускання[2]. Забезпечити це можливо, знайшовши правильний набір коефіцієнтів передавальної функції. Тобто стоїть проблема оптимізації коефіцієнтів передавальної функції цифрового фільтра із частотною вибіркою.

З графіків на рисунку 1 видно, що амплітуда (A) змінюється на всій області визначення частоти (f), причому без чіткої закономірності (часто зростає-спадає). Тобто маємо багато екстремумів на відносно малій області визначення. Крім того вигляд передавальної функції цифрового фільтра з частотною вибіркою є відносно складним. Один з можливих прикладів[3] :

$$H(z) = \sum_{n=0}^{N-1} h(n) z^{-n} = \sum_{n=0}^{N-1} \left[\frac{1}{N} \sum_{k=0}^{N-1} H(k) r^n e^{2\pi i n k / N} \right] z^{-n}$$

Класичним методам оптимізації важко працювати в таких умовах, хоч і існують безліч модифікацій. Наприклад, придумали стрибки по області визначення, аби вийти з локального екстремума.

Сукупність цих факторів спонукала розглянути не класичні методи оптимізації, а дещо інше. В дослідженні розглядаються еволюційні алгоритми, а саме – генетичний для пошуку глобального максимуму.

У генетичному алгоритмі використовуються такі біологічні поняття як популяція, хромосома, гени, схрещування, мутація. На початку кожної ітерації існує популяція хромосом. З популяцією роблять схрещування, генеруючи нащадків(в їх генах роблять мутацію для різноманіття популяції). Маючи батьків і дітей, за допомогою «фітнес-функції» відбирають найкращих особин для наступної ітерації.

Такий алгоритм не лякає ні кількість екстремумів ні складність функції. Він пристосований до роботи на глобальній області визначення з багатьма екстремумами. І що важливо, що «фітнес-функція» може бути значно простіша ніж функція, з якою йде робота.

Розроблене програмне забезпечення приймає на вхід наведені вище характеристики ЦФ (порядок, частоти зрізу, гарантоване затухання, нерівномірність). Використовуючи генетичний алгоритм знаходяться коефіцієнти передавальної функції. Для зручності буде вестися база даних. Аби з її допомогою можна було знайти коефіцієнти фільтра при заданих вхідних параметрах.

Список використаних джерел :

1. Рабинер Л., Гоулд. Теория и применение цифровой обработки сигналов.: Москва, 1978.
2. Голд Б., Рэйдер Ч. Цифровая обработка сигналов.: «Сов. Радио», 1973, 368 с.
3. Айфичер Э., Джервис Б. Цифровая обработка сигналов, Практический подход.: Второе издание, Москва, 2004.